



**Optimal UAS Assignments and Trajectories for
Persistent Surveillance and Data Collection from
a Wireless Sensor Network**

DISSERTATION

Nidal M. Jodeh, Lieutenant Colonel, USAF
AFIT-ENY-DS-15-S-062

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This is an academic work and should not be used to imply or infer actual mission capability or limitations.

AFIT-ENY-DS-15-S-062

OPTIMAL UAS ASSIGNMENTS AND TRAJECTORIES FOR PERSISTENT
SURVEILLANCE AND DATA COLLECTION FROM A WIRELESS SENSOR
NETWORK

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

Nidal M. Jodeh, B.S., M.A.S., M.S.
Lieutenant Colonel, USAF

26 December 2015

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED

OPTIMAL UAS ASSIGNMENTS AND TRAJECTORIES FOR PERSISTENT
SURVEILLANCE AND DATA COLLECTION FROM A WIRELESS SENSOR
NETWORK

Nidal M. Jodeh, B.S., M.A.S., M.S.
Lieutenant Colonel, USAF

Committee Membership:

Richard G. Cobb, PhD
Chairman

David R. Jacques, PhD
Member

Lt Col Jeremy S. Agte, PhD
Member

Adedji B. Badiru, PhD
Dean, Graduate School of Engineering and Management

Abstract

In this research, a method is developed for multiple unmanned aircraft assigned to fly optimal trajectories to survey and collect a pre-specified amount of data from a fixed, ground-based wireless sensor network. Assigning UAS to network nodes and flying optimal trajectories minimizes an objective weighting of flight time, distance, and control inputs. Individual sensor nodes are initially clustered by minimizing Euclidean distances to centroids by a k -means++ clustering algorithm. Next, a continuous time optimal control problem is formulated to solve for the optimal trajectory, minimizing a weighted sum of the time and control effort needed to collect sensor data. This problem formulation is a modified traveling salesman problem with neighborhoods and the control policy enables the airborne vehicle to change altitude, heading and velocity to best communicate with each ground sensor via omni-directional antennas. The optimal trajectory is solved for by direct orthogonal collocation methods also known as a pseudospectral method. Each unmanned aircraft is then tasked to fly the optimal trajectory within its assigned cluster, survey, and collect the minimum amount of required data from each assigned sensor. Sensor nodes with the most data are identified as high activity sensors. These high activity sensors are then reassigned to dedicated UAS by an expectation maximization of Gaussian mixtures method. UAS are assigned to new clusters and re-tasked with updated optimal trajectories. The new assignments and optimal flight trajectories are calculated to ensure increased revisit rates over the high activity sensors, while maintaining or decreasing time over sensors with relatively less activity. The optimal trajectories were simulated in software and hardware-in-the-loop environments and were shown to decrease flight time and distance when compared to traditional path planning techniques. Finally,

successful proof of concept flight tests were used to validate the unmanned aircraft optimal trajectory and data collection architecture. For the cases considered, a 14% to 32% reduction in flight time and distance was achieved when compared to traditional flight planning methods.

For my Wife and our children.

Acknowledgements

“If I have seen further, it is by standing on the shoulders of giants”

– Sir Issac Newton

Earning a PhD is not an endeavor one completes alone. I need to begin by recognizing my advisor, Dr. Cobb, whose encouragement, guidance, and wisdom were immeasurable over the last three years. The morning coffee runs were especially helpful, and not just for the short research discussions and caffeine, but as a means to get to know one another better. Our weekly meetings usually resulted in 30 minutes of sound research discussion and 30 minutes of in-depth Seinfeldian discussion, solving the world’s most pressing problems, and helping to keep me sane.

Thanks to my committee Dr. Dave Jacques and Lt Col Jeremy Agte for your feedback and positive words of encouragement. Thank you to my sponsor Dr. Derek Kingston at AFRL/RQQA for the research topic, sponsorship, support network at RQQA, and integrating my research into the flight tests. I am especially thankful to the crack flight test team of Steve Rasmussen, Dr. Dave Gross, Jeff Hill, and Brian Bean. From Muscatatuck to Rockhampton and all the test ranges in-between, you are masters of your craft. Thank you to Rick Patton and taking time to inspire my boys in the ANT Center.

Thank you to my peers, namely Tim Coon, Ted Masternak, Clay Humphreys, Ryan Carr, Gary Goff, and especially Riley Livermore. I could not have crossed the finish line without your coding and mathematical genius, insight, suggestions, and contributions. Thank you to my penthouse neighbors Jason Bindewald, Dave Meier, and to Tim Carbino, who graciously taught me the fine art of AFIT survival. I couldn’t have asked for a better set of classmates than the AFIT PhD Class of 2015.

Thank you to Kara Greene and Angie Suplisson, my pillars of support, always there for me, guiding me, hearing me out, checking up on my family, fixing my code, and teaching me LaTeX. I would not have completed the degree without you.

Thank you to my parents for instilling the value of lifelong pursuit of education and knowledge. Thank you to my mother for being the master chef and master caretaker. Thank you to my father. For everything I do well, it is because you taught me how, and for everything I do not do well, it is because I did not listen well enough to your advice. Stay strong Hajj, we'll get through this thing together. Thank you to my brother and sisters for your belief in me. Please stay strong for our parents.

Thank you to my bedrock, my foundation, my love, my wife; you have been the most patient of all, and for that I am forever in your debt. Thank you to my little family for your unending patience. You are the three brightest stars in my sky, guiding me home.

Praise be to God, for whom nothing is possible without. Alhamdulillah.

Nidal M. Jodeh

Table of Contents

| | Page |
|--|------|
| Abstract | iv |
| Acknowledgements | vii |
| List of Figures | xiii |
| List of Tables | xvi |
| List of Abbreviations | xvii |
| I. Introduction | 1 |
| 1.1 Motivation | 2 |
| 1.2 Background | 2 |
| 1.2.1 Wireless Sensor Networks | 2 |
| 1.2.2 Mobile Data Collection Agents | 3 |
| 1.2.3 Assignment Methods | 4 |
| 1.2.4 Optimal Trajectory Planning | 4 |
| 1.2.5 Mission Scenario | 5 |
| 1.3 Problem | 6 |
| 1.3.1 Dissertation Scenario | 6 |
| 1.3.2 Research Questions | 7 |
| 1.3.3 Research Goals | 7 |
| 1.3.4 Research Scope | 8 |
| 1.4 Assumptions and Limitations | 8 |
| 1.4.1 Sensor Modeling | 8 |
| 1.4.2 UAS Modeling | 9 |
| 1.4.3 Communications Modeling | 9 |
| 1.4.4 Optimal Routing | 10 |
| 1.4.5 Assignment Methods | 10 |
| 1.4.6 Command and Control | 10 |
| 1.5 Contributions | 11 |
| 1.6 Document Outline | 11 |
| II. Literature Review | 13 |
| 2.1 Terminology | 13 |
| 2.2 Unmanned Aircraft System | 15 |
| 2.3 Wireless Sensor Networks | 16 |
| 2.3.1 Sensors | 17 |
| 2.3.2 Wireless Sensor Network Architecture | 18 |
| 2.3.3 WSN Applications | 19 |

| | | |
|-------|--|----|
| 2.3.4 | WSN Limitations - Data Delivery, Energy Management, and Network Lifetime | 19 |
| 2.4 | Wireless Sensor Networks and Mobile Agents | 21 |
| 2.4.1 | Delay Tolerant WSN and Mobile Agents | 21 |
| 2.4.2 | WSN and UAS Applications | 24 |
| 2.4.3 | Data Latency | 25 |
| 2.4.4 | Data Latency Leads to Data Buffer Overload | 27 |
| 2.4.5 | Extend WSN Lifetime by Mobile Agent | 27 |
| 2.4.6 | Communication Protocol of WSN and UAS | 29 |
| 2.5 | The Shortest Path Problem | 30 |
| 2.5.1 | Traveling Salesman Problem | 31 |
| 2.5.2 | Vehicle Routing Problem | 33 |
| 2.5.3 | Solving the TSP | 34 |
| 2.5.4 | Additional TSP Constraints | 36 |
| 2.6 | UAS Assignment Algorithms | 36 |
| 2.6.1 | k -means++ | 38 |
| 2.6.2 | Gaussian Mixtures | 39 |
| 2.6.3 | Expectation Maximization | 40 |
| 2.7 | Trajectory Optimization | 41 |
| 2.7.1 | Trajectory Optimization - Direct Collocation Methods | 42 |
| 2.7.2 | Direct Orthogonal Collocation Solution Method | 43 |
| 2.8 | Summary | 44 |
| III. | Methodology | 46 |
| 3.1 | Approach Method | 46 |
| 3.2 | Initial Sensor to UAS Assignment | 46 |
| 3.3 | Initial Guess | 48 |
| 3.3.1 | Initial Guess by Traveling Salesman Problem Solution | 49 |
| 3.4 | The Optimal Control Problem | 51 |
| 3.4.1 | UAS Equations of Motion | 52 |
| 3.4.2 | Communications Model | 54 |
| 3.4.3 | Trajectory Constraints | 60 |
| 3.5 | Optimal Control Problem Formulation | 63 |
| 3.6 | Gaussian Mixture Model | 65 |
| 3.7 | Expectation Maximization | 65 |
| 3.8 | Reference Scenarios | 66 |
| 3.9 | Summary | 67 |

| | |
|--|-----|
| IV. Results | 69 |
| 4.1 Problem Setup, Settings, Values, and Software | 69 |
| 4.1.1 UAS State and Control Limits | 69 |
| 4.1.2 Software | 70 |
| 4.2 Sensor Clustering | 71 |
| 4.3 Initial Guess of the Optimal Trajectory | 71 |
| 4.4 Optimal Trajectory | 73 |
| 4.4.1 Optimization Software - GPOPS II | 74 |
| 4.4.2 GPOPS-II Settings | 75 |
| 4.4.3 Cost Function Weights | 75 |
| 4.4.4 Single Sensor Coplanar with One UAS | 77 |
| 4.4.5 Six Sensors on Two Planes with One UAS | 79 |
| 4.4.6 Sixteen Gridded Sensors with One UAS | 80 |
| 4.4.7 Ten Sensors with One UAS and No-Fly Zone | 84 |
| 4.4.8 Ten Sensors with Two UAS | 86 |
| 4.5 Sensor Reassignment for Ten Sensor Two UAS Scenario | 86 |
| 4.5.1 900 kbit Requirement for Sensors #6 and #10 | 88 |
| 4.6 Simulation | 98 |
| 4.6.1 Simulation Procedure | 99 |
| 4.6.2 Simulation Results and Conclusions | 101 |
| 4.7 Chapter Summary | 102 |
| V. Flight Test | 104 |
| 5.1 Introduction | 104 |
| 5.2 Purpose | 104 |
| 5.3 Equipment and Range | 104 |
| 5.3.1 Aircraft | 105 |
| 5.3.2 Autopilot and Control Stations | 105 |
| 5.3.3 Ground Sensors | 105 |
| 5.3.4 Test Ranges | 106 |
| 5.4 Experimental Procedure | 108 |
| 5.5 Settings and Values | 108 |
| 5.6 Results | 109 |
| 5.6.1 Data Collected | 110 |
| 5.7 Day Three Flight Time and Distances Flown Comparison | 112 |
| 5.8 Day Three Communications Analysis | 114 |
| 5.8.1 Day Three, Sensor Three Communciations Analysis | 115 |
| 5.9 Post Flight Discussion | 116 |
| 5.9.1 Communications Anomolies | 117 |
| 5.10 Recommendations | 119 |

| | Page |
|---|------|
| 5.11 Conclusion | 121 |
| VI. Conclusions and Recommendations | 122 |
| 6.1 Conclusions..... | 122 |
| 6.2 Contributions | 124 |
| 6.3 Potential Future Research | 125 |
| 6.4 Summary | 126 |
| A. Conference Paper #1 (January 2014) | 129 |
| B. Conference Paper #2 (January 2015) | 141 |
| C. Conference Paper #3 (January 2015) | 156 |
| Bibliography | 168 |
| Vita | 179 |

List of Figures

| Figure | Page |
|--------|---|
| 1 | DoD UAS Group Categories 17 |
| 2 | Notional WSN Based On Unit Disk Graph Model 18 |
| 3 | WSN Taxonomy Diagram 23 |
| 4 | Packet Reception Probability Versus Transmission Radius 30 |
| 5 | Classical TSP Example 32 |
| 6 | TSP Example with Dynamics 33 |
| 7 | TSP Example with Neighborhoods and Dynamics 34 |
| 8 | TSP for Dubins' Paths 36 |
| 9 | Flow Chart for UAS Data Collection in WSN 47 |
| 10 | Initial Guess Methods 51 |
| 11 | UAS States and Inertial Frame Definition 54 |
| 12 | Communications Model 59 |
| 13 | Heading Correction Loop 63 |
| 14 | Reference Scenarios 67 |
| 15 | 2D Ten Random Sensor Scenario Split Into Two Clusters 71 |
| 16 | Two Clusters of Sensors with TSP Initial Guess 73 |
| 17 | Optimal UAS Trajectories for Single Sensor Example 78 |
| 18 | Data Rate for Single Sensor Example 79 |
| 19 | UAS Controls for Single Sensor Example 79 |
| 20 | 3D, Six Sensor Optimal Trajectory 81 |
| 21 | Sixteen Sensor Gridded Layout 82 |

| Figure | | Page |
|--------|--|------|
| 22 | Sixteen Sensor Gridded Layout with One UAS..... | 82 |
| 23 | Re-planned optimal paths for two UAS given updated sensor areas of high activity. | 83 |
| 24 | Optimal UAS trajectory for 10 random sensors with no-fly zone and initial guess. | 84 |
| 25 | Ten Sensor Example with No-Fly Zones | 85 |
| 26 | Ten Sensor Reference Scenario with Initial Guess and Optimal Trajectory | 87 |
| 27 | Control and Data Rates - Ten Sensor Reference Scenario | 88 |
| 28 | Ten Sensor Reference Scenario After Cluster Update | 89 |
| 29 | Expection Maximization of Gaussian Mixture Model for 1, 5, and 10 Weight Factor on Sensors#6 and #10 | 90 |
| 30 | Ten Sensor Reference Scenario Comparison for Increased Data from Sensors #6 and #10 | 93 |
| 31 | Control and Data Rates Comparison for Ten Sensor Reference Scenario on the Left Hand Cluster for the Corners Scenario | 94 |
| 32 | Control and Data Rates Comparison for Ten Sensor Reference Scenario on the right Hand Cluster for the Corners Scenario | 95 |
| 33 | Ten Sensor Reference Scenario Data Collection Comparison Bar Chart | 96 |
| 34 | TSP and Optimal Trajectory Comparison in AMASE..... | 101 |
| 35 | Expection Maximization of Gaussian Mixture Model for 1-6 Weight Factor On High Activity Sensors #9 and #10 | 103 |
| 36 | Flight Test Control and Base Stations | 106 |
| 37 | AFRL WRSI Ground Sensor | 107 |

| Figure | | Page |
|--------|---|------|
| 38 | Wilmington Air Park | 107 |
| 39 | Numerical Trajectory (GPOPS-II) Solution for Wilmington | 110 |
| 40 | Flight Test Data Collected - Days One and Two | 111 |
| 41 | Numerical Trajectory (GPOPS-II) Solution for Wilmington | 112 |
| 42 | Flight Test Data Collected - Day Three | 113 |
| 43 | Numerical (GPOPS-II) Solution and Flight Test Trajectories | 116 |
| 44 | Sensor #3 Comm Messages Received at UAS | 117 |
| 45 | Message Receive Rate and Distance to Sensor | 118 |

List of Tables

| Table | | Page |
|-------|--|------|
| 1 | Synonomous Terminology from Various Related Literature. | 15 |
| 2 | UAS State and Control Limits. | 70 |
| 3 | Results of Various Initial Guess Methods. | 72 |
| 4 | GPOPS-II Settings | 76 |
| 5 | Ten Sensor Refernce Scenario Comparison of Flight Time and Distance (Left Cluster) | 92 |
| 6 | Ten Sensor Refernce Scenario Comparison of Flight Time and Distance (Right Cluster) | 92 |
| 7 | Selected Setting for Flight Test | 109 |
| 8 | Flight Test Day Three Telemetry | 114 |
| 9 | Comparison of Numerical and Flight Test Flight Times | 114 |
| 10 | Comparison of Numerical and Flight Test Distances Flown | 115 |

List of Abbreviations

| Abbreviation | Page |
|--------------|---|
| DoD | Department of Defense 1 |
| UAS | Unmanned Aircraft Systems 1 |
| ISR | Intelligence, Surveillance, and Reconnaissance 1 |
| WSN | Wireless Sensor Networks 2 |
| AFRL | Air Force Research Laboratory 5 |
| RPV | remotely piloted vehicles 13 |
| UAV | unmanned ariel vehicle 13 |
| R/C | radio controlled aircraft 13 |
| MA | Mobile Agent 14 |
| UGS | unattended ground sensors 14 |
| EUGS/E-UGS | Expendable UGS 14 |
| FAA | Federal Aviation Administration 16 |
| TSP | Traveling Salesman Problem 22 |
| TSPn | Traveling Salesman with Neighborhoods Problem 28 |
| VRP | Vehicle Routing Problem 33 |
| GMM | Gaussian Mixture Model 38 |
| EM | Expectation Maximization 38 |
| EM | Expectation Maximization 40 |
| NLP | Non-Linear Program 42 |
| GPOPS-II | General Purpose Pseudospectral Optimal Control Software-II 70 |
| AMASE | Air Vehicle Test Analysis and Simulation Multi-Agent Simulation Environment 70 |

| Abbreviation | | Page |
|--------------|--|------|
| CMASI | C ommon M ission A utomation S ervices I nterface | 98 |
| XML | Extensible Markup Language | 98 |
| WSRI | Wright State Research Institute | 104 |
| AFIT | Air Force Institute of Technology | 179 |

OPTIMAL UAS ASSIGNMENTS AND TRAJECTORIES FOR PERSISTENT SURVEILLANCE AND DATA COLLECTION FROM A WIRELESS SENSOR NETWORK

I. Introduction

THE Department of Defense (DoD) estimates manpower cost is the largest component in the operation of Unmanned Aircraft Systems (UAS). From planning, controlling, supervising, analyzing, re-planning, delivering data, and other functions, the human operator currently bares the majority of the burden for these tasks [1]. The most critical phases of mission profiles are often either manually performed or pre-programmed by human operators. These functions “include critical flight operations, navigation, takeoff and landing of unmanned aircraft, and recognition of lost communications requiring implementation of return-to-base procedures” [1]. Furthermore, UAS that conduct Intelligence, Surveillance, and Reconnaissance (ISR) missions often collect and deliver raw data. For example, live streaming video from a UAS requires human interpretation and analysis before it can be used for decision making. Ideally, future unmanned systems will be capable of many of these tasks autonomously, moving from pre-programmed to situational, decision-based maneuvers and from simply delivering raw data to directly providing actionable intelligence [1]. The DoD also predicts, “autonomous systems may even optimize behavior in a goal-directed manner in unforeseen situations (i.e., in a given situation, the autonomous system finds the optimal solution)” [1]. However, the UAS itself is limited by its on-board capabilities. Networking the UAS with standoff components, such as other UAS, manned vehicles,

and remote sensing create a force multiplying effect, dramatically improving a single UAS’s capabilities.

1.1 Motivation

To meet the DoD’s vision for future UAS capabilities and lower manpower costs, systems incorporating UAS and standoff components are needed. Rapidly maturing hardware require novel algorithms to enable UAS to autonomously collect and analyze data, make decisions based on collected data, and deliver pertinent information to the end users when their data are efficiently collected. Advanced Wireless Sensor Networks (WSN) can provide time critical and precise localized environmental information critical to decision making. Researching how to smartly combine multiple UAS, WSN, and algorithmic components motivates this research. To put the research goals in perspective, a brief background is given to assist in scoping and providing context to the research accomplished herein.

1.2 Background

1.2.1 Wireless Sensor Networks.

The concepts of wireless sensor networks, combined with efficient data collection, and network energy management¹ collectively compose a relatively new field of study, only introduced in the last two decades. The term “Wireless Sensor Network” began to appear in literature in the late 1990’s, specifically in Pottie in 1998 [2]. Immediately, the inherent problems of data collection and network energy management were apparent and became a focus of this burgeoning research area [2–4]. Early work focused on node-to-node multi-hopping data routing protocols within the network. However, as WSNs grew and matured, new applications began to appear such

¹A generic term used to describe various practices to extend the life of network components.

as forest fire detection and monitoring, environmental mapping, traffic monitoring and tracking, and in the military, battlefield surveillance [3]. These new applications further complicated data collection and network energy management as well as introducing newer challenges like network deployment, size, cost, upgrades, usable lifespan, survivability, and adaptability to changing network requirements.

Alternative data collection methods reduced network energy consumption rates and collected more data by using robots, a.k.a. mobile collection agents, to physically visit and survey sensor nodes within the network, collect their data, and deliver or relay the data back to the user or data sink. The mobile agents were assumed to have larger storage capability which in some architectures freed the WSN of the multi-hopping protocol, often the largest consumer of network energy reserves. Some of the earliest authors to propose the use of mobile agents as an alternative to the multi-hopping routing schemes were Tong et al. in 2003 [5], Zhao and Ammar in 2003 [4], and Moore et al. in 2005 [6].

1.2.2 Mobile Data Collection Agents.

The introduction of a mobile collection agent solved some problems in network data collection and energy management, however it opened a Pandora's box of new challenges. The relatively new field of WSN now intersected with the very mature fields of applied mathematics and combinatorial optimization. Two problems emerge specifically, assigning mobile agents to sensor nodes and finding optimal routes to visit a collection of nodes while satisfying specified criteria and obeying constraints. The more famous of these routing problems are the Traveling Salesman and Vehicle Routing Problem. The "best" route for a mobile agent traversing a WSN can only be defined by the users. For some, the best routes may be predicated on the collection of required data from each node in the WSN, or the best route maybe the short-

est distance required to visit each node, and for others the best route may change dynamically as the mobile agent travels the network.

1.2.3 Assignment Methods.

Given multiple UAS operating in a WSN, a method of task assignment is often desirable. In other words, to which UAS should each sensor in the WSN be assigned? Assignment methods can also have added benefit of deconflicting UAS flight paths. In this research, the problem is straight forward: divide the WSN into sub-groupings or clusters equivalent to the number of available UAS and assign UAS to respective clusters. The problem of grouping data sets, observations, or in this case ground sensors into like groups is known as “cluster analysis” [7]. Clustering enables the partitioning of data into meaningful subgroups or “clusters.” For this dissertation, the clustering of data is synonymous with clustering of the sensors when an exclusive, deterministic solution is desired. The assignment of tasks, targets, or geographical areas to UAS can also be accomplished by several methods. For example, consensus-based task allocations or auctioning methods, where UAS bid on particular tasks have been employed [8–10]. A common research focus in assignment methodologies is whether to use centralized or decentralized task assignment. Bakker argues that centralized task assignment methods are vulnerable to a single point of failure [11].

1.2.4 Optimal Trajectory Planning.

Calculating optimal routes for mobile agents in a WSN is still a relatively new field of study and research is ongoing. However, methods for mobile agent route planning are scarce. In 2005, Moore et al. considered a UAV locating clusters of sensors, but the routing was not developed as optimal [6]. In 2010, Li et al. surveyed 48 papers concentrating on sink mobility in WSNs [12]. Many, if not all, of the papers

reviewed by Li concentrated on energy management of the individual nodes of the WSN. Several authors considered portions of the work contained in this dissertation, but none have demonstrated a concerted effort to connect the methods to solve the broader problem of assigning and re-assigning airborne mobile agents to conduct persistent surveillance and data collection from a WSN.

1.2.5 Mission Scenario.

The scenario which served as a reference mission in this research was adopted from work at the Air Force Research Laboratory (AFRL) by Kingston [13], Marshall et al. [14], and Klesh et al. [15], who considered a UAS tasked to collect data from a sparse but tactically placed, wireless road sensor network. AFRL examined employment of WSN as a security apparatus, used to track intruders into vital areas or remote austere positions, such as Forward Operating Bases. The WSN employs static, ground-based wireless sensor nodes designed to collect road and foot traffic data within their individual sensing ranges, store, and then report that data to the overflying UAS when in range.

AFRL's proposed scenario is divided into four phases: patrol, isolation, delivery, and response. During the patrol phase, the UAS is tasked to visit all the sensors within the WSN, collect their data and determining if any regions of the WSN are experiencing relatively increased activity. If a region(s) of the WSN are in fact reporting increased activity, the isolation phase is triggered. Given this increased activity, the UAS is tasked to autonomously prioritize the region(s) of the WSN experiencing the increased activity. The goal of the isolation phase is to track and image the increased activity with electro-optical sensors onboard the UAS. Once the activity has been imaged by the UAS, the delivery phase ensures the appropriate agencies are provided the data. Finally, the response phase is triggered when the images are

analyzed by the user and the proper course of action is taken. This dissertation will formulate and solve the patrol and isolation phases of AFRL's proposed scenario as described below.

1.3 Problem

This research addresses the challenge of providing persistent surveillance and data collection by autonomously assigning and planning optimal UAS flight trajectories based on decision making information obtained from a static, ground-based WSN. In a wide, geographically dispersed WSN, the most efficient grouping and route of flight to visit each sensor node may not be entirely obvious. This problem is partially difficult since conditions across the WSN are often not uniform, making flight planning and data collection a dynamic process.

1.3.1 Dissertation Scenario.

Specifically, the WSN is subdivided into clusters of sensors and UAS are assigned to each cluster. Within assigned clusters, each UAS will calculate the optimal flight path minimizing time and control usage, subject to aircraft dynamics, airspace limitations, communication rates, and a minimum data requirement from each sensor. After initial orbits of the WSN, sensors reporting higher rates of activity relative to other sensors are regrouped into new clusters and the UAS are then reassigned. This enables the UAS to concentrate on sensors within the WSN experiencing the highest rates of activity while maintaining a minimal visitation rate to sensors with less activity. Given the communication range of the UAS, data may be reported back to the end user or data sink immediately, after the initial orbit, or after a series of subsequent orbits. Optimal flight planning ultimately leads to reduced fuel consumption, shorter flight times, shorter surveillance distances, and shorter response distances to

other areas of the WSN. Furthermore, having more relevant information from the WSN lends itself to being more actionable as opposed to raw data from the entire network.

1.3.2 Research Questions.

Based on the motivating sections and the problem scenario, specific research questions are formulated.

1. How should sensors be optimally allocated to more than one UAS?
2. How should an optimal control problem be formulated to solve for an optimal trajectory given the problem constraints?
3. As the problem evolves and data is collected, new information is provided. How should this information be incorporated into the problem in a dynamic manner?
4. How should the data collected be used to identify areas of interest in the WSN and can that data be used to dynamically weight each sensor for subsequent trajectory calculations?

1.3.3 Research Goals.

1. Given a WSN and multiple UAS, develop an optimal sensor to UAS assignment algorithm.
2. Given a WSN and UAS, develop an optimal control method to ensure sufficient data collection from all sensors, in the least amount of time and with minimal control effort.
3. Conditioned on the data collected as the problem evolves, develop a method to increase the visitation rates of the highest activity sensors and decrease or maintain the visitation of the least active sensors.

1.3.4 Research Scope.

This research will be limited to the UAS optimal assignment and control problem. There are many challenges associated with the concept of aircraft collecting data from a network of wireless ground-based sensors, namely path following (inner-loop control), collision avoidance, WSN energy management, data latency, data buffer overload, communications optimization, sensor localization, or maximum communication rate localization to name a few. While these areas of research are necessary and critical to furthering the WSN-UAS concept, they are beyond the scope of this research as many of these challenges are thoroughly addressed in other studies.

1.4 Assumptions and Limitations

To sufficiently bound the research tasks, several assumptions and limitations must be made about the sensors, UAS, communications environment, and hardware to constrain the scope of the problem and keep it tractable.

1.4.1 Sensor Modeling.

The WSN is assumed homogeneous. In other words, the transmission power, storage capacity, communication capability, and sensing capability of each sensor are uniform across the WSN. Sensor locations are known to the UAS a priori and do not move once emplaced. The sensors are also assumed to have sufficient energy reserves to last significantly longer than the duration of an entire UAS mission or stated objective. The sensors are assumed to be low powered, communication range limited, and often expendable. Sensors cannot communicate with each other due to the distance and communication protocols between each one. They are incapable of communicating via satellite due to power and bandwidth limitations. The overflying

UAS is assumed the only means of communicating sensor data. Finally, sensor storage capacity or data buffering is considered sufficient for the duration of the scenarios.

1.4.2 UAS Modeling.

The UAS is modeled as a point mass system. The UAS size, weight, and performance specifications are based on Group 1 (Small UAS) parameters as defined by the DoD. Transition from one flight condition to another is not considered as many of these vehicles can reach their maximum speeds and commanded altitudes in negligible amounts of time. It is also assumed the UAS has unlimited on-board data storage and range for the duration of the test scenarios. Stochastic disturbance inputs such as variable winds and other random effects are not considered. Performance limits on heading, pitch, and velocity are imposed. Control inputs are the time rate of change of heading, pitch, and velocity. Processing of data, calculating optimal paths, command, and control is assumed to be done at a central ground-based control station accessible by all UAS and monitored by the end users. UAS are considered airborne over a WSN at the beginning of any scenario.

1.4.3 Communications Modeling.

The UAS and each sensor is equipped with a single omni-directional antenna that is attenuated only by free-space spreading loss. The sensor transmission assumed is continuous and isotropic over the entire field consistent with a fixed transmitted power. No geographic limitations or azimuth and elevation requirements are imposed between the UAS and the sensor. Two-way communication and error checking are not considered. The UAS antenna is required to be in contact with each sensor long enough to ensure transmission of a fixed data package size. The multichannel equipped UAS is capable of simultaneously communicating with multiple sensors

when in range. Multiple UAS do not communicate with each other. Further details on the communications modeling are presented in Chapter III.

1.4.4 Optimal Routing.

For purposes of optimal routing, the trajectory cost between sensor nodes is considered equal in either direction. In other words, the solution flight paths are considered symmetric in either direction between any two sensors. Only one orbit of the WSN is calculated. Additional orbits are presumed to be multiples of the single orbit and the number of additional orbits is subject to user requirements, UAS limitations, etc. If dynamically tasked, optimal trajectories will be updated.

1.4.5 Assignment Methods.

In scenarios where multiple UAS are available, it is assumed that sensors are exclusively assigned only to one UAS, resulting in a conflict free solution. Although these scenarios may be easily run from a single central command and control computer, it is foreseeable that all the UAS can share the same algorithms and concur on similar solutions. However, for this dissertation, execution of the assignment algorithms are done from a centralized location. Finally, the UAS are not aware of the other UAS tasks, actions, or locations. If properly assigned, flight paths do not intersect, and thus are inherently deconflicted. Data from initial UAS orbits of the WSN is aggregated at the centralized location and any necessary reassignment is completed. There is no assumed reason that UAS are aware of the other's actions.

1.4.6 Command and Control.

The methods and solutions presented herein are theoretical. Any implementation of these methods is considered research only. Therefore, it is assumed the command

and control of the UAS and final destination for their collected data is a ground-based computer control station and the UAS do not necessarily communicate with each other. However, future operational command and control is foreseen on any of the UAS in coordination with a ground-based, human-supervised control terminal.

1.5 Contributions

Several specific contributions to the existing body of knowledge come from this research answering how to accomplish the patrol and isolation phase to achieve persistent surveillance and data collection.

1. A method and demonstration of a sensor to UAS assignment algorithm.
2. A demonstration of optimal control method to ensure sufficient data collection from all sensors, in the least amount of time and control effort.
3. A method to perform dynamic weighting of the sensors during a data collection mission and re-calculate an updated optimal trajectory based on the updated weighting.

1.6 Document Outline

This dissertation contains six chapters. Each chapter closely follows the general process used to solve the overall problem. The reader should recognize three major topic areas in Chapters II, III, and IV as sensor to UAS assignment, then UAS trajectory optimization, and sensor to UAS re-assignment. This first chapter introduced and motivated the problem along with assumptions, limitations, research questions, and outlined the approach. Chapter II reviews the relevant UAS-WSN terminology, various architectures, related challenges, and solution methods. Chapter III introduces the mathematical methodology and provides a framework for answering the

research questions. Chapter IV demonstrates, analyzes, and discusses the methodology and results by way of examples. Chapter V reviews flight test results based on the optimal flight path algorithm. Finally, Chapter VI lists the relevant contributions, future work, recommendations, and conclusion. Three appendices include publications written as portions of this dissertation.

II. Literature Review

The combination of WSN and UAS introduces a wealth of challenging problems. This chapter provides background on the common terminology, various WSN architectures, the objectives and solution methods of similar studies, and identifies areas where further research can be made.

2.1 Terminology

Across a wide spectrum of related studies, various terms are used to describe similar research areas as the one addressed by this research. In the most general terms, this dissertation explores methods to employ multiple UAS to fly and collect data from static, unattended ground sensors. The UAS analyze the data and are autonomously re-assigned to visit the ground sensors with the most activity. The importance of this terminology is made clear when conducting a literature search. Initially, research that seemingly appears unrelated is suddenly closely connected when the reader is aware of the various synonymous terms.

- **Vehicle** - Historically, UAS have been called *drones*, *remotely piloted vehicles (RPV)*, *unmanned ariel vehicle (UAV)*, *models*, and *radio controlled aircraft (R/C)* [16]. However the term UAS more accurately represents all the components required for an unmanned aircraft to operate, including support operations, ground stations, and operators [1, 16, 17]. In addition to the historical terms for UAS, other names also refer to the data collection and carrying entity performed in this and similar studies. For example, *mobile elements* [18–20], *mobile robots* [21, 22], *robomote* [23], *data mules* [22], and *ferries* or *data ferries* [24, 25]. The term *data mule* was itself defined as an acronym where “*mule*” stood for **M**obile **U**biquitous **L**AN **E**xtensions [26]. Undoubtedly these terms

are used to generalize the problem description, largely referring to any entity capable of acting in this data collection, transport, and delivery capacity, such as airborne, marine, or ground robots [21]. **In this research, the term Mobile Agent (MA) refers to any land, sea, or airborne collection robots. The term “UAS” only refers to the airborne mobile collection robot.**

- ***Ground Sensors*** - Mobile, static, unattended, and in some cases expendable, ground sensors also have many synonyms like *widespread sensors* [24], *static sensors* [21], *notes* [22], *nodes* [19, 24, 25], *sensor nodes* [18], *unattended ground sensors (UGS)* [27], and *Expendable UGS (EUGS/E-UGS)*¹ to name a few. **In this research, the terms “sensors”, “nodes”, and “sources” are synonymous and used interchangeably.**
- ***Wireless Sensor Network*** - The most common term referring to a network of widespread, immobile, unattended ground sensors is a *wireless sensor network* [18, 20, 28]. In some studies the sensors are mobile, but still collectively referred to as a WSN. **In this research, all sensors in a network are collectively referred to as a “WSN.”**
- ***Data Collection from Wireless Sensor Network*** - Finally, various authors refer to the action of collecting data from remote sensors by a mobile vehicle as *data harvesting* [19, 20, 22], *data exfiltration* [6], *data gathering* [18], and *data collection* [28, 29]. Although not widely used, some authors have formally defined it as the *Data Gathering Problem (DGP)* [22, 30]. Earlier, Data Ferry was used as a noun to describe the vehicle itself, while here data ferry is used as an adjective to describe the type of mission performed by the vehicle.

¹A product manufactured by Applied Research Associates, Inc. (ARA), and can be found at <http://forcepro.ara.com/products/force-protection/expendable-unattended-ground-sensors>, accessed on April 1, 2014

In this research “data collection” will be used to describe the UAS function within the WSN.

A summary of the synonymous terminology is collected in Table 1.

Table 1. Synonomous Terminology from Various Related Literature.

| Mobile Agent | Sensor | Data Collection |
|------------------------------------|--|-------------------------------------|
| <i>Mobile Elements</i> [18–20] | <i>Wireless Sensor Networks</i> [18, 20, 28] | <i>Data Collection</i> [28, 29, 31] |
| <i>Mobile Agents</i> [28] | <i>Widespread Ground Sensors</i> [24] | <i>Data Exfiltration</i> [6] |
| <i>Mobile Sink</i> [32, 33] | <i>Static Sensors</i> [21] | <i>Data Extraction</i> [22] |
| <i>Mobile Robots</i> [21, 22] | <i>Sensor Fields</i> [22] | <i>Data Gathering</i> [18, 34, 35] |
| <i>Mobile Carriers</i> [22] | <i>Motes</i> [22] | <i>Data Harvesting</i> [19, 20, 22] |
| <i>Mobile Nodes</i> [36] | <i>Nodes</i> [19, 24, 25] | <i>Data Ferrying</i> [25, 36] |
| <i>Mobile Data Collector</i> [37] | <i>Sensors</i> [28, 38, 39] | <i>Data Retrieving</i> [40] |
| <i>UAV</i> [16, 40, 41] | <i>Sources</i> [38] | |
| <i>Data Ferry</i> [24, 36, 42, 43] | <i>Unattended Ground Sensors</i> [27] | |
| <i>Data Mules</i> [22, 26, 44, 45] | <i>Expendable UGS</i> [46] | |
| <i>Robomotes</i> [23] | <i>Sparse Sensor Networks</i> [30] | |
| <i>Robots</i> [31] | <i>Ground Sensor Arrays</i> [28] | |
| <i>Aerial Relay Nodes</i> [47] | | |

2.2 Unmanned Aircraft System

The UAS has experienced an explosion in growth over the past decade. The DoD alone reported a meager 10,000 UAS flight hours in 2000, to over 550,000 flight hours in 2010 [48]. So much so that in the United States, regulatory and legislative

bodies have begun to regulate their use and operations in earnest [1]. While the primary users remain the DoD and academic institutions, commercial users are only at the door step of a multi-billion dollar industry. In 2014, the Federal Aviation Administration (FAA) cited a study by the Teal Group who predicted total research, development, and procurement will rise from \$5.2 billion to \$11.6 billion annually from 2014 to 2024. Furthermore, the Teal Group expects total spending on UAS world wide to top \$89.5 billion over the next ten years [49]. The FAA underscores the small UAS (under 55 pounds) as an economically viable instrument attractive to commercial users. If appropriate legislation and regulatory guidelines are in place, roughly 7,500 commercial UAS are expected to be in use by 2019 [49]. This figure is in addition to private, military, and academic UAS users.

The DoD categorizes UAS into five groups, listed in Figure 1. DoD defined Groups 1 and 2 are comparable to the FAA’s definition of small UAS. Although the UAS in this research is modeled after Groups 1 and 2 type vehicles, the methods and algorithms presented can easily be applied to any of the UAS Groups. Groups 1 and 2 type vehicle definitions were chosen for use in this research for several reasons. They are the most prolific among research communities and the most commonly used vehicles for flight test, especially for research and development in academia. Group 1 and 2 vehicles lend themselves nicely to the mission scenario introduced earlier in sections 1.2.5 and 1.3.1. They are also the type of vehicles used by AFRL for its flight test and demonstration of the methods developed in this research.

2.3 Wireless Sensor Networks

The following subsections cover the various aspects of WSN, such as the sensors, architecture, applications, and limitations.



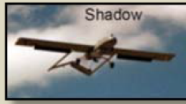


| UAS Groups | Maximum Weight (lbs) (MGTOW) | Normal Operating Altitude (ft) | Speed (kts) | Representative UAS | |
|------------|------------------------------|--------------------------------|--------------|---|---|
| Group 1 | 0 – 20 | <1200 AGL | 100 | Raven (RQ-11), WASP |  |
| Group 2 | 21 – 55 | <3500 AGL | < 250 | ScanEagle |  |
| Group 3 | < 1320 | < FL 180 | | Shadow (RQ-7B), Tier II / STUAS |  |
| Group 4 | >1320 | | Any Airspeed | Fire Scout (MQ-8B, RQ-8B), Predator (MQ-1A/B), Sky Warrior ERMP (MQ-1C) |  |
| Group 5 | | > FL 180 | | Reaper (MQ-9A), Global Hawk (RQ-4), BAMS (RQ-4N) |  |

Figure 1. DoD groupings of UAS by size and performance [48].

2.3.1 Sensors.

Individual sensors, or *sources*, may be small in size (from a few cubic centimeters to a few thousand cubic centimeters), independent of other sensors, and capable of communicating wirelessly to neighboring sensors or mobile data collection agents. Individual sensor nodes may be expendable, camouflaged, and placed very far from the WSN hub or *sink*. A single sensor can collect, measure, record, and report environmental changes within its immediate vicinity. Sensors can measure distance, direction, vibrations, motion, and so on [12]. Data is recorded as electro-optical, doppler, seismic, magnetic or infrared information with a corresponding time stamp [13]. For example, ground vehicles passing by a sensor may be recorded as seismic disturbances, triggering a video camera, time-stamped, and transmitted to the overflying UAS. Generally, sensors are considered low powered, low cost devices with limited

battery life. Communications range limited, sensor nodes are spatially constrained if required to communicate directly with each other. Similarly, communicating with overhead satellites is not practical due to limited power and lack of a directional antenna.

2.3.2 Wireless Sensor Network Architecture.

WSN are typically a collection of low powered, remote sensor nodes, or *sources*, and a central collection node, or *sink*. One type of WSN architecture relies on a sensor-to-sensor data delivery, or multi-hopping communications scheme. Data collected at the sensor nodes can be transmitted by forward multi-hopping along various sensors on a path toward the sink [50]. Multi-hop schemes often uses a two-dimensional unit disk graph model to represent a sensor's communication range [12]. For example, two sensors are connected if their disks intersect as seen in Figure 2. A WSN may

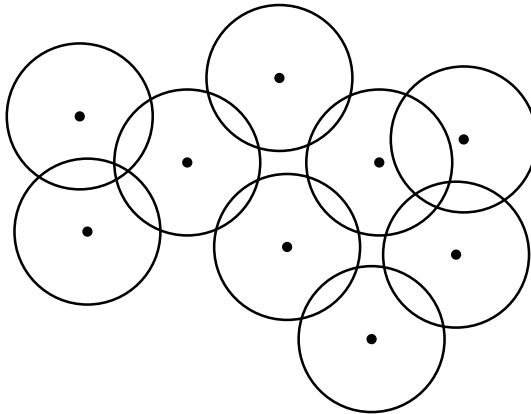


Figure 2. Notional WSN based on unit disk graph model [12].

contain any number of sensors, more if a larger physical area is being covered for example. In some WSN architectures, sensors rely on a fixed, ground-based central collection node, the sink, within proximity to the sensor, to collect and transmit vital information to end users [51, 52]. This sink requires line of sight with the sensor

nodes, a communication link with operators, power, and physical protection. In some WSN, sinks are more costly than sensors and not considered expendable.

2.3.3 WSN Applications.

WSN are currently employed in a variety of applications such as monitoring seabird habitats [53], coral reefs and fisheries [54], structural health [55], forest fires [51], fertilizer and pesticide applications in agriculture [56], vehicular ground traffic in urban areas [57], border patrols [58], and tracking illegal border crossings². Some current and expected applications include security awareness, disaster response, search and rescue support, and critical infrastructure monitoring at power facilities, ports and pipelines [16]. Kingston et al. and Spillings et al. also describe a Base Defense problem where WSN are used to provide security for military bases, forward operating bases, and remote towns and villages [59, 60]. Motivated by a constant threat on operational bases in recent conflicts, Spillings et al. describes an urgent need for an integrated suite of sensors and countermeasures to enhance a defensive perimeter [60].

2.3.4 WSN Limitations - Data Delivery, Energy Management, and Network Lifetime.

There are significant limitations to using a WSN. Limitations to data delivery methods and energy management schemes contribute to shorter network lifetime and are the subject of the following research.

A multi-hop architecture can be efficient in delivering data to the central collection node and is favored in applications where data is required near instantaneously and delays are not tolerated. While multi-hopping from sources to sinks works well in networks with consistently available power, in more austere or hostile environments,

²Case study cited by Applied Research Associates on Jan 30, 2014 and can be found at <http://forcepro.ara.com/case-study-california-rancher-deploys-e-ugs-seismic-sensors-intercept-illegal-immigrants>, accessed on Aug 22, 2015

the remote sources survive on their own. Remote and unreachable sources are expendable at the end of their useful life. Further, multi-hopping schemes limit the size of WSN as sensor nodes must be within radio range of each other to relay data from the outermost sensors.

Although multi-hopping data is desirable because of the speed of delivery, energy consumption is high and depleted non-uniformly, especially at the sensors closest to the sink [61–64]. Gu and Castano both identified the increased energy consumption of sensors in close proximity to the sink [52, 65]. The closer a sensor is to the sink, it is widely accepted that sensor is required to relay more data, while the outermost sensors only transmit once to the next nearest sensor in the relay chain. If the sensors closest to the sink are consuming more energy than the further ones, they are more likely to be depleted of energy ahead of the others. Castano solved this challenge by using multiple sinks and determining their optimal placement to ensure coverage of the WSN. This method is also subject to data losses, errors, and interference, especially with lengthiest transmissions and in dense WSN.

If only one sink is employed, a single point of failure is exposed and redundancies may be required. In austere, less secure, or remote locations, sinks may be difficult to place, maintain, and protect. In some scenarios, it may be advantageous to deploy sensors over a large geographical area, sensing specified locations, and be independent of a central sink. The use of a centralized sink tasked with collecting all the data from remote sensors often demands its own power sources. In the case of the low powered sources, line of site with the sink is required. Assuming a hostile environment, the sink may also require security.

Multi-hopping routing schemes work well with unobstructed line of sight, non-hostile environments with stable energy supply and accessibility. However, multi-hopping becomes increasingly difficult in the absence of the aforementioned condi-

tions. In some cases, a mobile sink or mobile agent is employed to collect and deliver data from the sensors as an alternative to the centralized, immobile sink. Mobile agents are capable of reaching each sensor source, eliminating the need for data hopping from the sensors to the sink. Mobile agents enable the fixed sinks and sensors to be free of communication network requirements between each other and placed in high interest area where increased monitoring is needed [40].

It has clearly been shown that WSN can operate successfully but are limited by inherent challenges. Alternate WSN architectures utilize a mobile agent to collect data from sensor nodes and minimize or eliminate the need to relay data via multi-hop. The balance of this chapter is dedicated to the WSN and mobile agent problem, with attention paid to the WSN-UAS type problems.

2.4 Wireless Sensor Networks and Mobile Agents

Only in the last 10 years has extensive research on the combination of mobile agents and WSN been accomplished. The mobile agent and WSN each bring their own set of benefits and challenges. Integrating the mobile agent and WSN solves some existing challenges, introduces new challenges, and complicates others. The main challenges of WSN and mobile agent integration include data latency, network energy consumption, and path planning. The following subsections discuss the various challenges and their solutions.

2.4.1 Delay Tolerant WSN and Mobile Agents.

Li et al. diagrammed a taxonomy of various WSN and mobile agent solution approaches [12]. In Figure 3, Li categorized WSN into two main groups, delay tolerant networks and real-time networks. As the name indicates, delay tolerant networks are WSN where the relative age of the data, when it arrives at the sink or delivered

to the user, is still useful minutes, hours or days after the data has been collected. This is most commonly referred to as the data latency problem. The delay tolerant network is better suited with mobile collection agents, where sensor data is collected and stored on the mobile agent for later delivery to the sink or end user. The use of a mobile agent in delay tolerant networks also alleviate the WSN of the traditional energy consuming sensor to sensor multi-hopping communications methods.

Li specifies two prevailing approaches for data collection by mobile agents in delay tolerant networks: either make direct contact with the sensor or rendezvous with the data at some intermediate sink in the network. Under the direct contact branch, Li reviews stochastic, Traveling Salesman Problem (TSP), and label-covering solutions; under the rendezvous-based data collection, Li reviews fixed-track, tree-based, and clustering solution methods.

The second category Li describes, not depicted in Figure 3, is a real-time WSN where data is delivered nearly as soon as it is collected. This is a near-real time flow of data from source to the sink or end user in matters of seconds. In real time WSN, Li cites examples like forest fire monitoring or battlefield surveillance where time is of the essence in data delivery. The reader is directed to [12] for further information on these problems and their solution methods.

Under two realistic scenarios, this research could very well fall into either delay tolerant or real time WSN categories. If one were to assume the UAS, acting as the mobile sink, surveys and acts on data collected from the WSN, only to report the data at a later time, the architecture would be considered “delay-tolerant.” However, if the UAS, surveyed, collected, and immediately relayed the data back to end user via radio or satellite link, the UAS would be no more than a mobile non-collecting, relay node in the network.

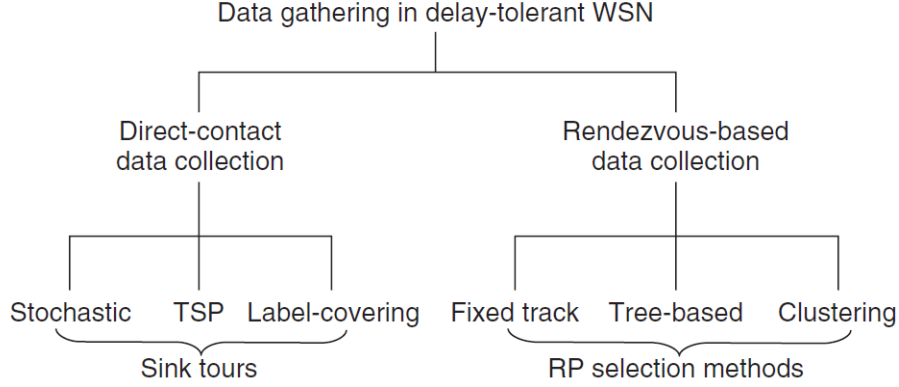


Figure 3. WSN Taxonomy Diagram [12].

Li et al. further specified two categories for mobile agents, “controlled” and “uncontrolled.” The controlled agent follows predetermined and predictable paths, and is under the control of the end user or a specific path planning algorithm. The uncontrolled mobile agent is one where the path is unknown and not subject to predetermined criteria or algorithms, and is best described as adding a sink to a wild animal for data collection in WSN in a forest or a taxi cab acting as a sink in a traffic monitoring WSN for example [12]. **Utilizing Li’s taxonomy, this research can be defined as data gathering in delay-tolerant WSN via a controlled mobile agent making direct contact with sensors with TSP-like solution approaches.**

In contrast to the multi-hop architecture, delay tolerant WSN nodes upload their data to the collection agent directly, such as a ground-based agent or UAS. Using a mobile collection agent enables a more spatially dispersed WSN, in turn making sensor placement a more strategic process, and especially useful when there is a limited quantity of sensors.

Mobile agents bring with them their own advantages and disadvantages. The first advantage, already mentioned above, enables sensors to be more spatially dispersed.

Second, energy consumption rates are relatively equal across the WSN as each sensor may only be required to communicate with the mobile agent, performing a single-hop communication. Thirdly, it is assumed that mobile agents have a larger data storage capability and can collect and store more data than any single sensor. Fourth, it is assumed that some mobile agents are capable of relaying data from individual sensors directly to the data sink. There are also disadvantages in using mobile agents, such as in scenarios where data is not relayed directly to the sink, and the sensor's data now resides on the mobile agent and is unusable until processed at the sink. If the mobile agent must deliver its data to the sink, a delay is incurred until the data is delivered. In a delay tolerant WSN, this condition is often accepted and of minimal impact to the overall mission of the WSN. Mobile agents can also be limited by logistics. For example, the need for energy, maintenance, and spare parts is inevitable for any mobile robot. Also, a ground-based mobile collection agent may be limited by terrain and an airborne collection agent limited by weather.

2.4.2 WSN and UAS Applications.

Alternative architectures replace the fixed sink(s) with a mobile agent, where the mobile agent visits all the sensors to collect and relay their data to necessary users. Specific types of mobile agents, like the UAS, enables the WSN to span many kilometers of area, limited only by range of the UAS. In this case the UAS is ideally suited to collect data from a widely dispersed WSN, that would otherwise be beyond the communications range of a ground-based central collection node. This architecture enables the UAS to address a large number of widely distributed sensors as opposed to a fixed sink addressing each sensor within a limited communications range. However, employing a UAS has trade-offs. These include weather limitations, airspace and

communication frequency management requirements, UAS logistics and maintenance, as well as processing and response time [13].

Researchers are applying the WSN-UAS architecture to solve a wide variety of problems. Abdulaal et al. considered the use of mobile sensors to track flood levels and movement. The mobile sensors are deployed from UAS which continue to track them after deployment [66]. Faical et al. used UAS to disperse pesticides onto crops and a WSN to monitor the dispersant. Feedback from the WSN enable the UAS to alter its flight path to compensate for winds or missed areas of the crop [56]. Fargeas et al. developed methods for a UAS to pursue an intruder into a WSN in 2015 [67].

2.4.3 Data Latency.

If a mobile agent is tasked to visit each sensor, the multi-hop architecture is reduced to a single hop between the sensor and mobile agent. This approach conserves sensor energy and reduces potential data loss, error, and interference. However, this method introduces data latency. The data, now residing on the mobile agent, must be delivered to a sink or end user, and is subject to the speed of the mobile agent. As the mobile agent visits more sensors along its route, data from the first sensors has the greatest latency period by the time it is delivered to the sink. To minimize latency, several authors combined multi-hopping and used a mobile agent. WSN are subdivided into sub-groups called, “clusters,” and data from each cluster was transmitted to one sensor within each cluster, known as “caching” sensors. The mobile agent would now only visit the caching sensors, collecting the aggregated data, and saving travel time and distance. This is widely referred to as a hybrid or rendezvous approach; effectively the data and mobile agent meet half way [68–70]. The authors of [18, 19, 47, 70, 71] addressed data-latency as a primary goal of their research.

Almi'ani uses iterative tree search methods to optimize the amount of forward hops from the originating sensor to the caching sensor subject to the mobile agent tour lengths [18]. Almi'ani considered two scenarios: 1) first the mobile agent is required to start and end at the same location, in other words return to the starting point to deposit the collected data, and 2) the mobile agent is free to start and stop anywhere in the wireless sensor network. The first is solved by a clustering based algorithm, where sensors are grouped in clusters around the caching sensors, and iteratively conducts a tree search for the best solutions that minimize the forward hops [18].

Moazzez-Estanini et al. minimized data latency by solving for a classical TSP solution and then employing a novel path splitting algorithm. This enabled the first Hamiltonian cycle solution to be split at key nodes into smaller nodes [19]. Chiu solved a variant of the with clustering-based genetic algorithm. The goal was to reduce the tour length for the data agent in order to minimize data latency [71]. By examining the communication ranges of all the sensors in a WSN, overlapping communication ranges were identified as intersections and established the waypoint for the data agent to visit. This eliminates the need to visit each sensor directly, as in the classical TSP, but now requires only a location common to a small cluster of sensors within proximity to each other instead. Chiu did not describe the mobile agent in detail. Rather, Chiu simply wanted to explore a new genetic algorithm to solve the TSPn problem. Zhang also considered a combination of algorithms that chooses the best combination of ground to ground (multi hop) and ground to air (via UAS) to deliver the data to sink [47]. Alqaralleh examined optimizing the multi-hop clustering solution, where the WSN is divided into clusters, and subsequent algorithms decide on the best sensor to cluster all the surrounding sensors' data. A ground-based mobile agent visits all the selected cluster sensors [70]. The author utilized heuristic based

solutions for partitioning the WSN and determining the clustering sensors within each partition.

Many of the papers reviewed make a critical assumption that data must be delivered back to the base station or sink and cannot be transmitted by radio, hence data latency is introduced and should be minimized. The method in this research does not make the same assumption and hence, alleviates the data latency issue. For this dissertation, the UAS, acting as the mobile agent, is not initially required to deliver the collected data to the sink, but rather to analyze it and autonomously make a decision on its next course of action. If necessary, the UAS is capable of transmitting data to end users by various means.

2.4.4 Data Latency Leads to Data Buffer Overload.

Another aspect of data latency is data loss. Gu assumed limited storage capacity on-board each sensor [20]. With limited storage capacity, the need arises for the sensor's data buffers to be downloaded with some regularity or at the very least, when they are reaching capacity. Gu considered the limitations of data buffers at each sensor and designed a partition g-based algorithm to schedule the movement of the mobile elements in wireless sensor networks such that there is no data loss due to buffer limitations or overflow [20]. This dissertation assumes data buffers have sufficient capacity for the duration of any scenarios and will not consider their limitations.

2.4.5 Extend WSN Lifetime by Mobile Agent.

The use of a mobile agent minimizes sensor energy consumption by visiting each sensor and reducing the multi-hop scheme to a single hop between the sensor and the mobile agent, resulting in a more uniform energy state across the entire WSN. How-

ever, in large scale WSN, it becomes impractical for a mobile agent to visit each sensor. Many proposals consider the hybrid approach mentioned earlier in Section 2.4.3

Many other studies found optimal trajectories for mobile elements visiting sensors networks, but used sensors energy consumption or data latency as motivating challenges to the problem. Similarly, this problem solves for an airborne mobile agent optimal trajectory. While this author acknowledges that the sensor's energy consumption is a necessary consideration, for this research, a sensor's energy is considered sufficient to complete the proposed scenarios (or missions).

Many studies explore the energy consumption required in the data transmission between the sensors, central collection node, and mobile agent [18, 32, 44, 47, 52]. A common solution involves a combination of single and multi-hops from distributed sensors to cache sensors and the use of mobile agents to collect data from the cache sensors.

Almi'ani et al. employs a hybrid of mobile elements and data multi-hopping with the goal of extending network lifetime [18]. The study minimized the path length of the mobile agent and the communication distance between sensor nodes, where sensors are clustered around a singular sink for the mobile agent to communicate with.

Tekdas demonstrated collecting data from ground-based sensors utilizing ground robots to solve a Traveling Salesman with Neighborhoods Problem (TSPn) . Tekdas also proved battery life is extended when ground sensors download data only to a passing robot. This is in contrast to constantly transmitting data to a central gateway node or long distances to data repeaters or other ground sensors.

Castano addressed the issue of WSN lifetime by developing a hybrid heuristic algorithm that group sensors to ensure consistent coverage of the targets of interest,

only activating those that are needed. The author introduces several base stations for data delivery and compares it to a single base station reference [52].

Zhang et al. examines a solution whereby the data can be routed to the base station or other sensors by either the UAS or sensor network. The author proposes optimizing a multi hop solution by either UAS to UAS, UAS to WSN, or WSN to base station [47].

Guo et al. considers a heuristic topology control algorithm whereby the energy state of each sensor is known and optimal combinations of neighborhoods with caching sensors is determined [63]. Neighborhoods and caching sensors may vary for each successive visit by the mobile agent. Guo specifically addressed non-uniform energy consumption by accounting for differing residual energy among the sensors. Using graph theory and optimizing the forward hopping paths from source sensors to caching sensors, Guo is able to manage the individual energy states [63].

2.4.6 Communication Protocol of WSN and UAS.

The quality of the communication signals is also an area of interest. Authors of [72, 73] proposed efficient communications protocols between the WSN and UAS. Ouyang et al. considered a ground-based mobile agent and a UAS capable of optimizing the beam shape to increase signal strength [74]. Jiang and Swindlehurst examined a method to dynamically adjust the UAS heading to maximize the upload data rates [75].

Kuruvila et al. assumed a fixed signal to noise ratio and using a unit disk graph model, developed a probability function based on the non-dimensional radial distance from the sensor. Figure 4 shows the probability of successful packet reception based on distance. Later, in Section 3.4.2.2, a model with different parameters but similar behavior is developed.

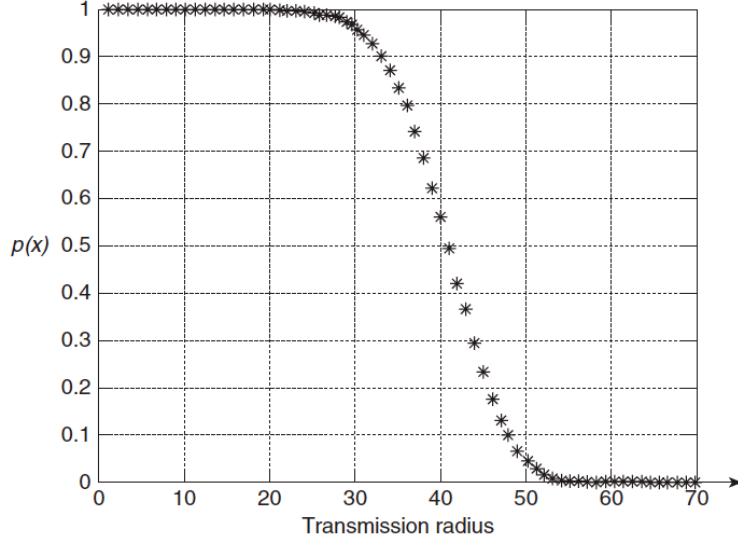


Figure 4. Packet reception probability versus transmission radius [76].

2.5 The Shortest Path Problem

In the most general of terms, the research considered in this dissertation is concerned with finding the best path or circuit necessary to visit all points within a set of points while satisfying a list of constraints. The best path is often defined as optimal if a set of criteria are met and the cost to visit all the points is minimized. In Applied Mathematics and theoretical computer science, one of the most classical problems is determining the shortest path required to visit a given set of points. This combinatorial optimization problem is NP-hard [77]. Often solved with graph theory, the problem has wide ranging applications from school bus routes, to mail delivery routes, to robotic arm paths in computer chip manufacturing. Finding an optimal route has the obvious benefits of saving time and fuel, finding the shortest distances, and maximizing other resources such as driver availability or limited vehicle storage capacity.

Problems related to the shortest path problem have been studied for nearly a century or more; these include the Traveling Salesman, Vehicle Routing, Canadian Traveler, and Chinese Postman problems [78, 79]. Each has its own unique set of constraints and definitions. Of the various path problems, the Traveling Salesman and Vehicle Routing problems are considered the closest to the research done in this work. This section will compare and contrast the problem definitions and solution methods. This research effectively solves a TSP problem, but using a considerably different and wholly novel method. Utilizing a mobile agent to visit fixed sensors in a WSN has similarities to the Traveling Salesman and the Vehicle Routing Problems and are discussed in the next section.

2.5.1 Traveling Salesman Problem.

The TSP has been studied extensively and asks the following question: Given a list of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? Figure 5 illustrates a simple TSP example where the mobile agent must visit all the nodes once and return to the base. When the distance of travel is an objective, the problem is referred to as the Euclidean Traveling Salesman Problem. Like the Euclidean Traveling Salesmen Problem, a mobile agent must visit all the ground sensors at least once in the shortest possible distance. When the traveling salesman is required to be only within proximity, or *neighborhood*, to the city (node) prior to moving to the next city, it is said to be a TSPn [80] and illustrated in Figure 7. Kingston and Macharet both state that finding the optimal path for data collection from a WSN is “very similar to the Traveling Salesman Problem with Neighborhoods [35].” Nesamony et al. developed a solution for a mobile agent using the TSPn model [81, 82].

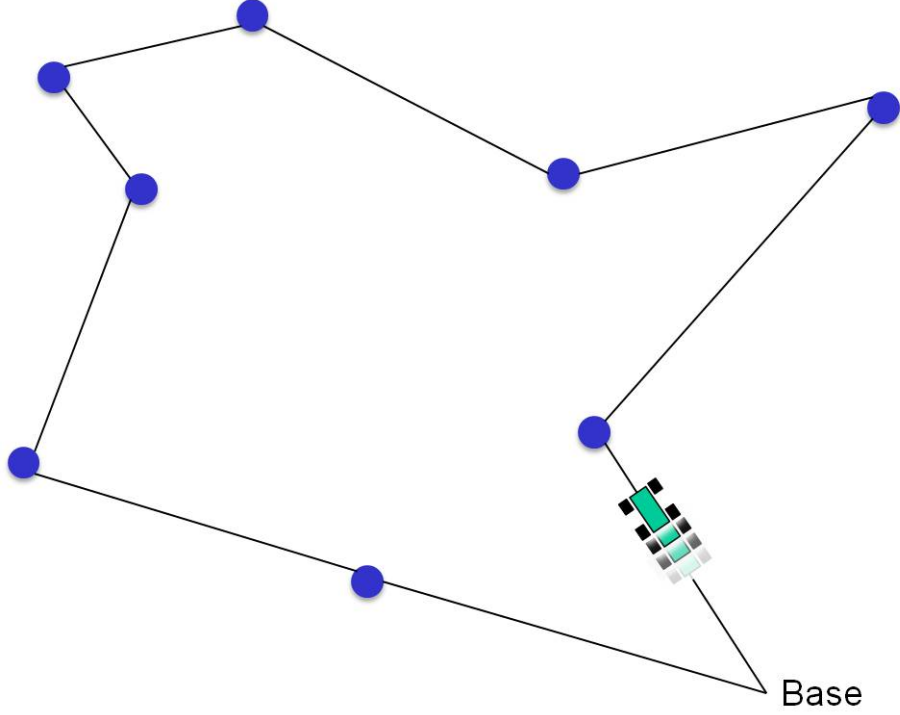


Figure 5. Classical TSP Example.

Like the TSP_n, the UAS in this research is required to be within radio range, hence the term “neighborhood,” of all the sensors at least once on its first orbit of the WSN. However, unlike the TSP_n, subsequent orbits of the WSN may require the UAS to visit certain sensors more than once or be well inside the neighborhood to collect more data (as discussed in Chapter III). Also, unlike the TSP_n, the common graph theory edge and vertex paths do not accurately represent the turning dynamics of the UAS, although several authors have adapted the TSP_n with turning dynamics [35, 45, 80]. A more accurate description would be TSP_n with dynamics as seen in Figures 6, 7, and 8. Although turning dynamics are incorporated into the TSP_n, their solutions do not necessarily apply to the data collection challenge in this research as the mobile agents are only required to “touch” the neighborhood of each node, as previously illustrated in Figure 7. These paths also do not meet the same optimality criteria specified later in this chapter and in Chapter III.

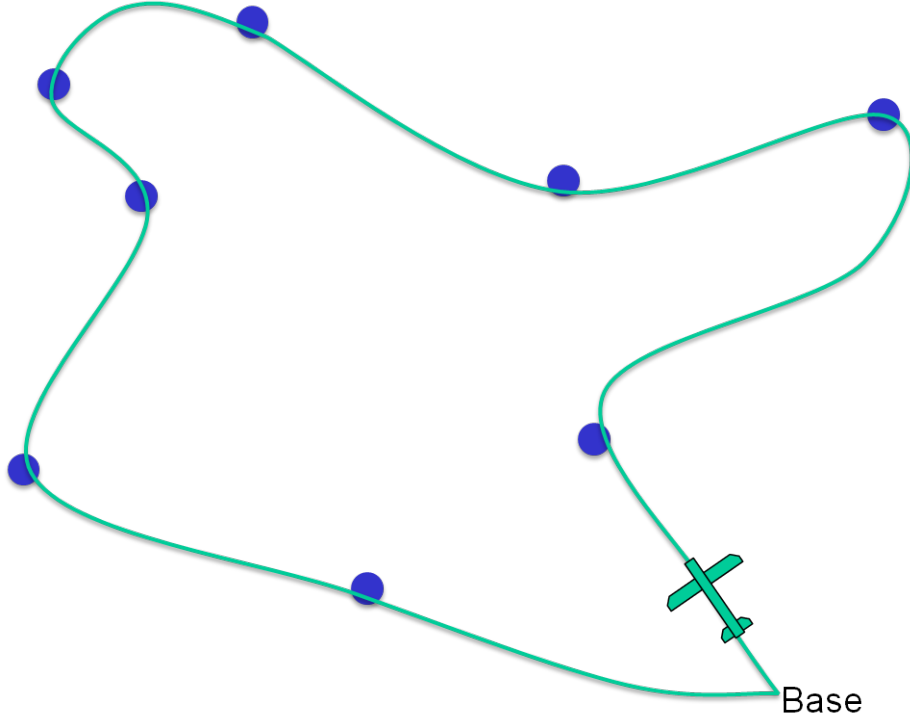


Figure 6. TSP Example with Dynamics.

continuous

2.5.2 Vehicle Routing Problem.

The Vehicle Routing Problem (VRP) is a more constrained version of the TSP. The VRP attempts to solve the logistical problem of delivery or collection of goods to and from customers and depots. Often, VRP's are associated with a fixed vehicle capacity, set number of available delivery vehicles and drivers, attempting to find the most efficient routes to deliver or collect goods. Common constraints in the VRP are road networks, time windows, available drivers, and delivery vehicle capacity. To compare the VRP to this research, the goods are considered to be the data, the customers to visit are the sensors, and the depot is the base station. In contrast to the VRP, the problems considered in this research are free of predetermined routes, drivers, fuel limits, and storage capacity among other limitations. This research will

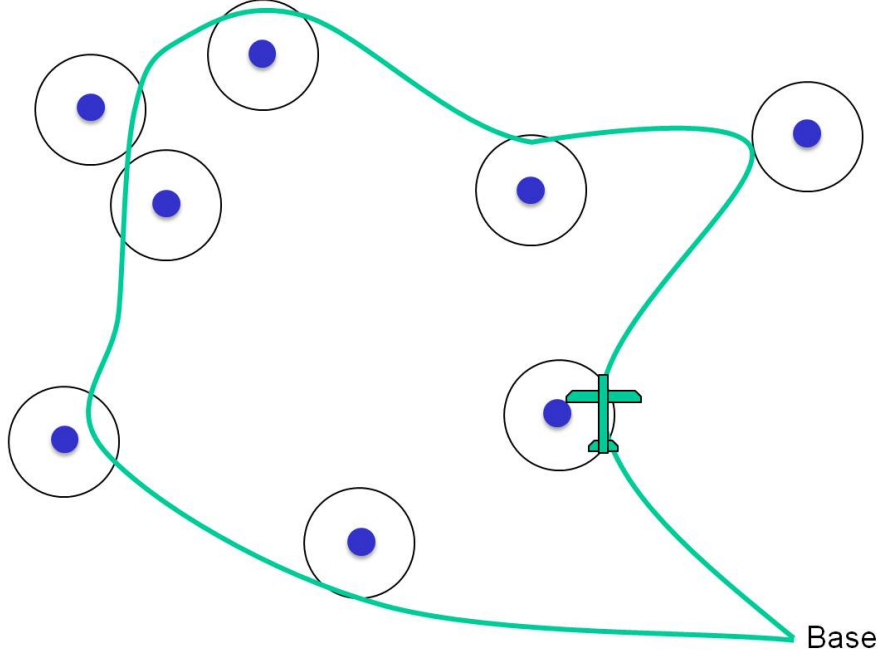


Figure 7. TSP Example with Neighborhoods and Dynamics.

attempt to handle various numbers of UAS, sensors, and collect large amounts of data within the optimization routines and it is assumed memory capacity and fuel are sufficient for the duration of all the scenarios.

2.5.3 Solving the TSP.

TSP and VRP problems are often solved using graph theory. Points and intersections on a graph are known as vertices and connections between them are known as edges or arcs [83]. Connecting the vertices and forming a path is referred to as a cycle of the graph, and is a subset of the set of edges and vertices that form a path such that the first node of the path corresponds to the last [84]. A cycle that uses each graph vertex of a graph exactly once is called a Hamiltonian cycle. A Hamiltonian cycle, also called a Hamiltonian circuit, is a graph cycle (i.e., closed loop) through a graph that visits each node exactly once [84]. The Hamiltonian cycle is a subset of the TSP problem in graph theory requiring the most efficient (i.e., least total distance)

Hamiltonian cycle a salesman can take through each of n cities. No general method of solution is known, and the problem is NP-hard in combinatorial optimization [84]. If the direction of visit from one sensor node to another is not specified, the path is also referred to as simple, regular, or more commonly, an undirected graph. Additionally, if the route in both directions between two nodes are weighted the same, or the cost of travel is the same in both directions, the TSP is commonly referred to as a symmetric TSP, and in turn halves the possible solutions. In contrast, if the cost of travel were different in the opposite direction, the problem would be referred to as an asymmetric TSP. The research proposed herein is defined as a symmetric and Euclidean TSPn. When the cost function satisfies the triangle inequality, we can design an approximate algorithm for the TSP that returns a tour whose cost is not more than twice the cost of an optimal tour. These tours are said to be a Metric TSP [85].

Wichmann et al. constructed a classical TSP solution to visit all the nodes in a WSN and included vehicle turning constraints to smooth out the path after the TSP solution was calculated [45]. Wichmann used graph theory and a Smooth Path Construction algorithm that first solves the TSP, then adds a minimum turn radius at the vertices, connecting them by edges and arcs. Macharet et al. defined an optimal path solution by solving the TSPn with a Dubins' vehicle model. The Dubins' model imposes the maximum possible curve (turn rate) and constant velocity assumptions [35]. Figure 8 illustrates a classical TSP solution and a TSP solution constrained with the Dubins' model [35]. Chatterjee et al. solved various symmetric TSP using genetic algorithms and generated results with 3.2% or better of a known optimum for given population sets [86]. Furthermore, Chatterjee et al. conclude that although their solutions were only near optimal, they are still quality inputs to algorithms that benefit from near optimal solutions.

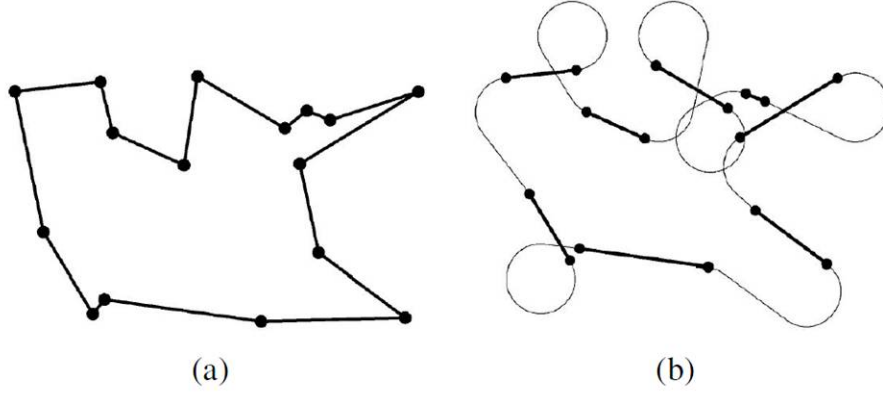


Figure 8. TSP path (left) and TSP for Dubins' Path (right) [35].

2.5.4 Additional TSP Constraints.

By leveraging concepts from the TSP and VRP, it is possible to consider additional constraints on this research. For example: 1. Time Windows - Data may be perishable or time sensitive and sensors must be visited within a certain time window, 2. Data Buffer Overload - Sensors may have limited data storage, and must be visited prior to loss of data due to buffer overload, 3. Order of Visit - It may be necessary to visit sensors in certain order, or 4. Sensor Energy - Sensors at the end of their usable life may need to be visited ahead of others with more energy reserves. Although relevant and challenging, these concepts are beyond the scope of this research.

2.6 UAS Assignment Algorithms

When multiple UAS are available, dividing the WSN into subgroups is advantageous. Many studies have examined the multiple UAS task assignment prob-

lem [9, 87, 88] for a variety of applications. However, most were concerned with dynamic task deconfliction and assignment for teams of UAS. One of the most common assumptions is that UAS are capable of communicating with each other [87, 88].

Bakker investigated a multi task Consensus-Based Auctioning Allocation Algorithm for multiple UAS. Bakker believes that centralized task allocation algorithms suffer from network congestion and possible conflicts if the current states of all the vehicles are not known at all times [11]. He also adds that a centralized auctioneer limits the potential of the UAS. Centralized control may also introduce a single point of failure. Bakker proposes that each UAS optimize its own path and share information when possible with nearby UAS in a bid to win the task. The UAS with the lowest cost bid will win the task. The ground sensors are assumed immobile as their data is collected, analyzed and compared to other ground sensor data. When adequate information is collected, the UAS can re-plan based on the sensors with the most ground activity [11].

Peng Wei et al. specifically addressed the sensor to UAS assignment problem [50]. Peng Wei et al. stated that single UAS data collection schemes are not robust as a failure of the UAS will result in no data collected. Furthermore a single UAS solution will undoubtedly consume more time as the single UAS is required to collect all the data [50]. To protect against the single UAS weaknesses, Peng Wei et al. propose a multi-UAS solution and a iterative balanced assignment with integer programming algorithm. Like many authors, Peng Wei also clustered sensors around a single caching sensor. Peng Wei’s objective was to minimize the total upload time.

Another approach is simply to sub-divide the sensors spatially, based on their geometric location. This sub-division is often called “cluster analysis.” Also described as the unsupervised classification of objects into similar but meaningful groupings [89].

Two common methods are the k -means++ and the multivariate Gaussian Mixture Model (GMM) optimized by the Expectation Maximization (EM).

2.6.1 k -means++.

Clustering takes advantage of the available UAS and reduces the time required to survey all the sensors. Using a Euclidean distance measure, subgroups or clusters are generated based on distances to a center of mass, or centroid. Initial development of the k -means approach (without the ++ designation) by Lloyd, employed a random starting point for all centroids simultaneously [90]. This initial seeding method leads to longer processing times and less than optimal solutions when compared to improved clustering methods. The initial seeding step simply selected all the cluster centroids uniformly at random from the available data. A perpendicular bisector method between the selected centroids was used to construct a Voronoi diagram. A new mean is determined in each section and the centroids are moved accordingly. The process repeats until a predetermined tolerance for the centroid movement is met. Lloyd's seminal algorithm is popular due to its simplicity and speed but limited in accuracy and is only guaranteed to find a local minimum [91].

The k -means++ algorithm is based on work by Arthur and Vassilvitskii [91] and seeks to smartly seed the initial centroid selection prior to executing the k -means method. Rather than selecting all the centroids initially, k -means++ selects one centroid uniformly at random with a high probability of minimizing the overall error. This step is then repeated for all the centroids. Arthur and Vassilvitskii state the k -means++ substantially outperforms k -means in terms of accuracy and speed [91].

k -means++ is ideally suited for spatial clustering problems when a distance metric is chosen. The algorithm treats each data point as a location in space, and partitions the clusters based on a distance measure [91]. This clustering method is ideally suited

to the spatial and geometric nature of the initial division of the WSN. This will be explored as part of this research.

2.6.2 Gaussian Mixtures.

Gaussian distribution models can also represent clusters of data [7, 92]. If the sensors are treated as a data set, a multivariate normal distribution is used to show their assignment to a cluster. Given a distance metric, the mean of each distribution represents the cluster centroid. Sub-groupings or clusters are often referred to as components and the data as observations. Multiple Gaussians of two or more variables are often referred to as a multivariate GMM, and used to represent the entire data set. GMM and subsequently the maximum likelihood of a data point residing in one Gaussian or another depends on the initial seeding. Initial seeding is easily accomplished by defining the initial means as the centroids from the k -means++ algorithm.

Both k -means++ and GMM with EM can produce “*hard*” or “*soft*” clustering. “*Hard*” clustering stipulates that data points must reside in only one Gaussian (cluster). This is also known as an “*exclusive*” assignment. In contrast, “*soft*” clustering permits data points to reside in multiple Gaussians, often along some common border, where posterior probability determines a relation between multiple Gaussian. For example, a posterior probability of 0.5, on a scale of 0 to 1, means a sensor has a 50/50 chance of belonging to either one cluster or another, also known as “*soft*” clustering. For this dissertation, individual sensors cannot simultaneously be assigned to more than one cluster at any given moment and are considered “*hard*” clustered.

2.6.3 Expectation Maximization.

When underlying data sets change in a GMM, updating and relocating the GMM's distributions is often solved by iterative methods, namely by Expectation Maximization (EM). EM can maximize the log-likelihood of observed data, simultaneously optimizing large numbers of variables, and able to find adequate estimates for unobserved data [93]. Generally, EM is an optimization method to estimate some unknown or unobserved parameters, given a set of observed or measured data [93, 94]. Unobserved parameters may be the mean and covariance of the unknown data set. EM iterates between two steps, Expectation and Maximization, to compute the maximum likelihood estimate in the presence of unobserved data. Maximum likelihood estimates model parameters for which the observed data is most likely [95]. In the Expectation step, unobserved data are estimated based on observed data. In the Maximization step, the likelihood function is maximized using the estimated data from the Expectation step. Convergence is guaranteed since likelihood increases with each step [95]. However, the EM algorithm is known to suffer from a slow rate of convergence, which occasionally gets stuck in local maximums, and is ill suited for problems with large numbers of clusters. The method breaks down when any cluster's covariance matrix becomes ill-conditioned (singular or nearly singular) [7, 93]. Previous works introduced the expectation and maximization steps [96], but Dempster et al. authored the seminal work that proved convergence and coined the name EM [97]. Moon applies the EM algorithm to signal processing, where data dropouts are considered unknown data, leading to estimation of a likely signal mean in presence of noise [98]. Multivariate GMM with EM is used in this research as a second sensor to UAS assignment methodology.

2.7 Trajectory Optimization

Most of the related work presented thus far considered some optimal path calculation, but they had differing motivating factors such as data latency or WSN energy consumption. Li reviewed several methods for route planning, including combinatorial solutions using graph theory, mixed integer linear programs, brute force, and learning-enforced methods. The closest work presented by Li involved data gathering in delay tolerant networks where the mobile agent collects data directly from the sensor node. From that work, Li identified two works that dealt with TSP solutions for data, however the constraints on visit order were minimal, often defaulting to remaining sensor energy reserves as a path planning constraint. In all the papers reviewed by Li, none specifically included aircraft dynamics, optimal control theory, or direct collocation methods.

In 2010, Cobano et al. considered aircraft dynamics and chose to use a partial filter to predict a flight path given various atmospheric conditions and other uncertainties. This path was designed to be robust, but not optimal. Cobano used a WSN as an example and he successfully collected data from each sensor node. Data collection was measured in number of messages received not data message size [40]. In 2008 and 2012, Nigam et al. considered control of multiple UAS for persistent surveillance but not with a WSN. Nigam's approach considered a more heuristic control policy and auction-based algorithm for multiple aircraft. In 2015 Fargeas et al. used a WSN with a UAS for intruder pursuit; however, this approach was designed to isolate a single intruder where as this research is designed to isolate a region [67]. In 2013, Kingston et al. found greater difficulty with tracking single intruders in WSN in practice than simply isolating a probability region of activity [99].

2.7.1 Trajectory Optimization - Direct Collocation Methods.

Hargraves was one of the earliest authors to use direct collocation to solve the optimal control problem for trajectory optimization [100]. Interest has increased in the use of Direct Collocation Methods to determine constrained optimal trajectories. Specifically, researchers have used Direct Collocation to solve problems for unmanned ground vehicles [101], UAS [28, 102–104], manned transport class aircraft [105], hypersonic vehicles [106], and spacecraft [107]. In 2009 Geiger used direct collocation methods for optimal route planning of UAS, but did not apply his solution to a WSN [104]. In 2009 Hurini et al. also used direct collocation methods for route planning, but demonstrated it on a ground-based vehicle.

Geiger et al. employed direct collocation to optimize the flight path of an unmanned aircraft. Geiger was able to flight test solutions as well. Geiger noted that a trade-off existed between extensive collocation points and the frequency of an updated solution, especially when incorporating external sensor data [104]. Although few collocation points may result in a sub-optimal solution, changes can be accounted for more efficiently, especially for dynamic trajectory updates [104]. Geiger also stated, “an accompanying reduction in the solution time horizon is required to maintain accurate dynamics interpolation [104].”

One paper considered an alternative to direct collocation. Horn et al. proposed a neural network to improve computational efficiency and scalability of the direct trajectory optimization methods. A method that uses neural networks to approximate the system dynamics, objective functions, and gradients was first presented and demonstrated in simulations. Horn says, “this method removes the need for collocation and thus reduces the Non-Linear Program (NLP) size [41].” The method has been extended to demonstrate its scalability to multiple-UAS problems and to problems involving a UAS with complex constraints [41].

2.7.2 Direct Orthogonal Collocation Solution Method.

Numerical methods to solve optimal control problems are generally categorized as indirect and direct methods. Betts conducts an extensive review of numerical methods to solve optimal control methods in [108]. Indirect methods are solved using calculus of variations and Pontryagin’s Maximum principle to determine the first order necessary conditions for optimality. The cost functional is first augmented with co-state and Lagrange variables. The first-order necessary conditions are determined by taking the first variation of the augmented cost function and setting the first free variable to zero. The necessary conditions form a Hamiltonian boundary-value problem (HBVP). The indirect method is highly accurate and insures the satisfaction of the first order optimality conditions. However indirect methods suffer from a small radii of convergence and require analytically solving for the HBVP. They also require an accurate initial guess of the co-state, and given path constraints, and a priori knowledge of the constrained and unconstrained switching structure [109–111].

In contrast, direct methods are transcribed directly to a nonlinear programming problem without deriving the first-order necessary conditions. They have a larger radii of convergence as compared to indirect methods and therefore do not need a good initial guess. Nor do they require an initial guess of the co-state or knowledge of switching structure *a priori*. However, the direct methods are susceptible to inaccuracies due to missing or inaccurate co-state information [110, 111]. Often, direct methods are said to solve “*directly*” for the control, rather than “*indirectly*” through the co-state term.

Direct orthogonal collocation methods have become increasingly popular to directly solve for optimal control problems solutions, as shown in [102, 104–106, 112–116]. In the aerospace field, direct orthogonal collocation methods are synonymous with Pseudospectral Methods [110, 113]. Numerous publications in the last decade

have thoroughly described and developed direct orthogonal collocation methods methods, namely seminal works by Rao, Benson, Huntington, and Ross [103, 107, 110, 111, 117]. Especially in the areas of aerospace trajectory optimization applications, the reader is directed to recent work by Geiger, Suplisson, Masternak, and Smith [102, 104–106]. Direct orthogonal collocation is explored as a means to solve for the UAS optimal trajectory problem.

2.8 Summary

Nearly all the aforementioned authors agree that for any data collection problem from a wireless sensor network by a mobile agent, the three major challenges include reducing data latency, WSN energy consumption, and optimal trajectory planning for the mobile agent. Many papers surveyed utilized graph theory to solve for a TSP like solution. Furthermore, most papers considered a hybrid forward multi-hop approach where sensors are clustered around a caching sensor and the mobile agent visits only that caching sensor, instead of each individual sensors. Few papers consider the communications model in the path planning modeling, and only one referenced a minimum data required criteria. None of the papers surveyed considered the collection of a minimum set of data from a WSN by a mobile agent using direct collocation methods. Although the optimal control and trajectory of a mobile agent was well studied, only a narrow scope of authors used the direct collocation methodology to solve the problem. None of the authors using direct collocation methods had considered a problem where a minimum data constraint from a WSN is required. As UAS are increasingly employed in WSN, the usefulness of optimal trajectories that minimize data latency, maximize WSN energy states, and collect data is clear. Furthermore, the work in this dissertation has practical and immediate application

to work by AFRL. Therefore, it is with a high degree of confidence that the proposed research herein is novel, relevant, and usefull.

III. Methodology

THIS CHAPTER presents the methodology used to solve the problem posed in Chapter I. The chapter is divided into several sections. The first section illustrates the flow of a typical scenario. The next section discusses the initial sensor to UAS assignment methodology. Then, the communications, UAS model, and trajectory constraints are explained followed by the optimal trajectory solution method. The final section describes the re-assignment procedure.

3.1 Approach Method

A simplified flow chart in Figure 9 illustrates the solution process. The process is initialized with the required information, namely the number of available UAS, sensor number and coordinates, and a minimum required data size. Once the initial information is input, the assignment and optimal trajectory algorithms compute their respective solutions. For changes in the problem characteristics such as number of available UAS, sensor locations, number of sensors, or data rates, the left side feedback loop of Figure 9 is completed. If the aforementioned characteristics remain unchanged for the duration of the problem, and only high activity sensors are identified, then the right side feedback loop of Figure 9 is completed.

3.2 Initial Sensor to UAS Assignment

This section describes implementation of the k -means++ clustering algorithm first introduced in Section 2.6.1. As stated in Chapter II, k -means++ is ideally suited for spatial clustering when a distance metric is chosen. The algorithm treats each data point as a location in space, and partitions the clusters based on a distance

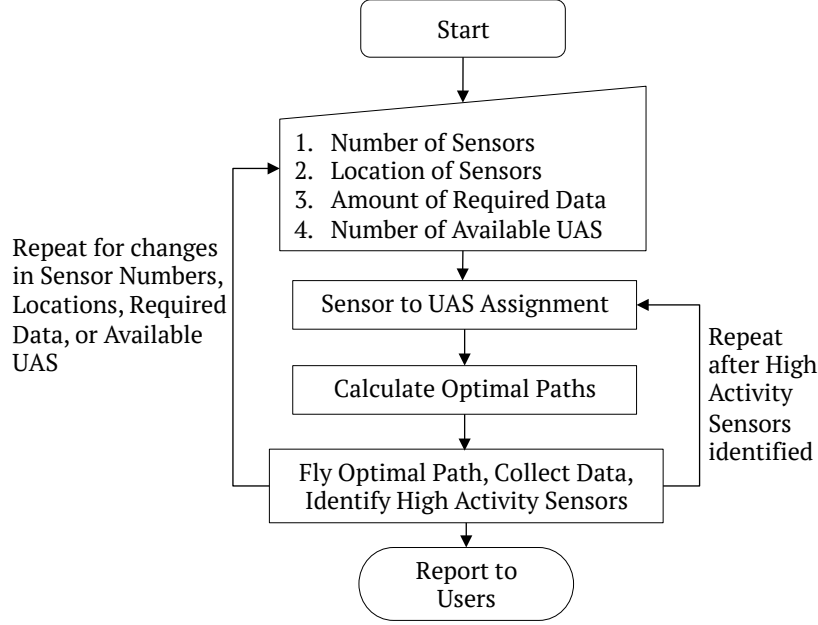


Figure 9. Flow chart for UAS data collection in WSN.

measure [91]. This clustering method is ideally suited to the spatial and geometric nature of the initial division of the WSN.

Given a set of N UAS, the individual sensors are grouped into N clusters. For Q sensors in \mathbb{R}^3 , define $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_j, \dots, \mathbf{s}_Q\}$ to be the set of three-dimensional cartesian coordinates of each sensor. Where the individual sensor coordinates are $\mathbf{s}_j = \{x_j, y_j, z_j\}$. For N total clusters required in \mathbb{R}^3 , defined as a set of three dimensional cartesian coordinates in the set $\mathbf{C} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k, \dots, \boldsymbol{\mu}_N\}$. The total number of required clusters, N , is also equivalent to the total number of UAS available. k -means++ finds a partition such that the sum of the squared error between the centroid (mean) $\boldsymbol{\mu}$ of the cluster and its assigned points is minimized [91, 118].

The squared error between the centroid and the cluster points is defined by

$$J(\mu_k) = \sum_{\mathbf{s}_j \in \mathbf{C}_k} \|\mathbf{s}_j - \mu_k\|^2 \quad (3.1)$$

However, k -means++ finds N centroids (means), \mathbf{C} , that seek to minimize the sum of the squared error over all N clusters,

$$J(\mathbf{C}) = \sum_{k=1}^N \sum_{\mathbf{s}_j \in \mathbf{C}_k} \|\mathbf{s}_j - \mathbf{C}_k\|^2 \quad (3.2)$$

Algorithm 1 enumerates the k -means++ method used in this research.

Algorithm 1 k -means++ Clustering Algorithm [91, 119]

Input: N available UAS and Q Sensor coordinates $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_j, \dots, \mathbf{s}_Q\}$

Output: N initial cluster centroid coordinates $\mathbf{C} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k, \dots, \boldsymbol{\mu}_N\}$ and Sensor Clusters $\mathbf{S} = \{\mathbf{s}_1^1, \dots, \mathbf{s}_j^k, \dots, \mathbf{s}_Q^N\}$.

- 1: Choose one point from \mathbf{S} uniformly at random as the initial centroid $\boldsymbol{\mu}_1$
 - 2: Calculate the minimum squared distance, D^2 , from each sensor coordinate to each centroid defined thus far by $D_i^2 = \min \|\mathbf{s}_j - \mathbf{C}_k\|_2^2$
 - 3: Choose the next center $\boldsymbol{\mu}_i$, by selecting $\boldsymbol{\mu}_i = \mathbf{s}' \in \mathbf{S}$ with probability $\frac{D(\mathbf{s}')^2}{\sum_{s \in \mathbf{S}} D(s)^2}$
 - 4: Repeat Step 3 until all N centers chosen
 - 5: For each $k \in 1, \dots, N$, set the cluster \mathbf{C} to be the set of points in \mathbf{S} that are closer to $\boldsymbol{\mu}_k$ than they are to $\boldsymbol{\mu}_l$ for all $l \neq k$
 - 6: For each $k \in 1, \dots, N$, set $\boldsymbol{\mu}_k$ to be the centroid of all points in \mathbf{C} : $\boldsymbol{\mu}_k = \frac{1}{|\mathbf{C}|} \sum_{s \in \mathbf{C}} s$
 - 7: Repeat steps 5 and 6 until \mathbf{C} no longer changes
-

3.3 Initial Guess

The trajectory optimization routine iterates on an initial guess towards an optimal solution. Providing a “good” initial guess helps the software to quickly converge on an optimal solution. The initial guess is also critical to the problem set up as the sensors were spaced far enough apart that insufficient gradient information was avail-

able for the optimization routine to adequately evaluate all feasible search directions. Although not zero, the gradient information from distant sensors is likely below the solver’s computational threshold. The initial guess of states and controls provides the first search direction, and is not subject to all the optimality constraints. Three methods were tested:

1. Two Point Initial Guess - This first method designates only the initial and final states and controls of the UAS at the first sensor.
2. Simply TSP - The second initial guess method employed a TSP solution for all the given sensors. This method included the first Two Point Initial Guess method, plus a direct sensor to sensor trajectory. This TSP solution is not an optimal trajectory, but ensured all the sensor locations are included in the initial guess. This method is more akin to “connecting-the-dots” and often violate the aircraft turn rate constraints.
3. Spline Fit TSP - Similar to the second initial guess method mentioned above, this final method smoothed out the direct trajectory TSP solution into a path that did not violate the turn rate constraints, making it more a more realistic flight path.

Figure 10 illustrates the three initial guess methods. The second and third initial guess methods, determined by solving the TSP, are expanded upon in the next section. This method of providing the location of all the sensors via the initial guess is consistent with the assumptions in Chapter I.

3.3.1 Initial Guess by Traveling Salesman Problem Solution.

The initial guess was formulated using a Euclidean symmetric TSP solution. Given a complete graph, $G = (V, E)$ with non-negative edge costs, find a minimum cost path

visiting every vertex, V , exactly once. The cost of each edge, E , is defined as the distance between any two sensors. In terms of this research, given a WSN, find the shortest path that visits every sensor exactly once and returns to the initial sensor. The problem is referred to as a Metric TSP if all path distances are symmetric and fulfill the triangle inequality,

$$dist(s_1, s_2) \leq dist(s_1, s_3) + dist(s_3, s_2), \quad \forall s \in \mathbf{S} \quad (3.3)$$

Further, if the vertices (sensor locations) correspond to points in a d -dimensional space and the cost function is based on a Euclidean distance, the problem is a special case of the Metric TSP, referred to as the Euclidean TSP. For purposes of this research, $d = 3$, and the Euclidean distance between any two sensor coordinates \mathbf{s}_i and \mathbf{s}_j is given as,

$$dist_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, \quad \forall i = 1 : Q, j = 1 : Q \quad (3.4)$$

This method results in a TSP solution, but with sharp corners at each sensor. Figure 10b illustrates a hypothetical four sensor scenario and a simple TSP solution. A potentially better guess involves smoothing the path as in the third initial guess method. To smooth the path, a cubic spline interpolating polynomial is fit to the simple TSP solution. For repeated, closed circuit orbits of the WSN, the first and last end point conditions must match to ensure a smooth and continuous path. Figure 10c illustrates the spline fit TSP initial guess. This is accomplished by matching the first and second derivatives at the end points [120]. The cubic spline interpolation was completed by using MATLAB's "csape" command in a function written by John D'Errico [120, 121].

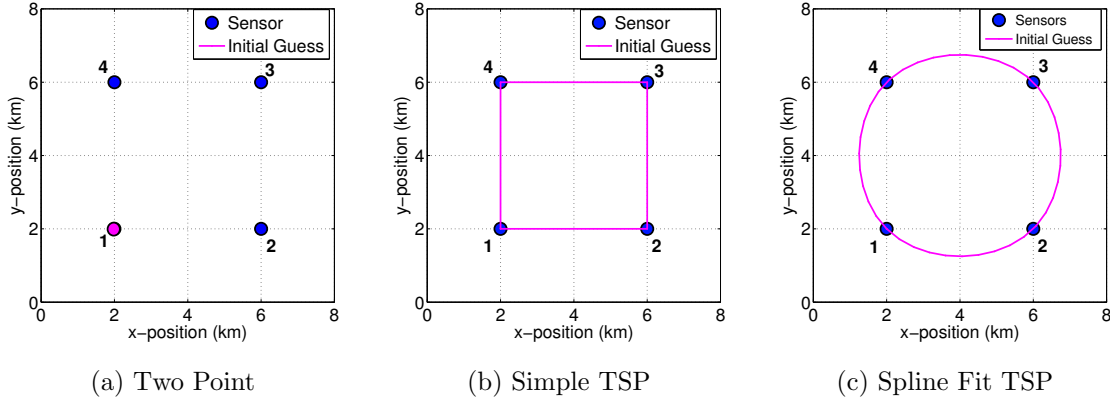


Figure 10. Three initial guess methods, (a) Two-point at sensor #1, (b) a simple TSP, and (c) a spline fit TSP solution.

3.4 The Optimal Control Problem

Simply stated,

“The objective of optimal control theory is to determine the control signals that will cause a process to satisfy the physical constraints and at the same time minimize (or maximize) some performance criterion” [109]

– Donald E. Kirk

In other words: find the optimal control inputs to the UAS, that satisfy the performance limits of the UAS and sensors, remain within the airspace boundaries, and minimize the flight time and control inputs, which result in the optimal flight trajectory. Formally, find the optimal control, \mathbf{u}^* , in the set of admissible controls, U , that causes the system dynamics, $\dot{x}(t) = f(x(t), u(t), t)$ to follow an optimal trajectory, \mathbf{x}^* , in the set of admissible trajectories, X , that minimizes the cost function, J , and satisfies the boundary and path constraints [109]. Conventionally stated by [103, 111, 112], and without loss of generality, determine the control $\mathbf{u} \in U$, state $\mathbf{x} \in X$, initial and final time, on the interval $t \in [t_o, t_f]$ that minimize the cost functional,

$$J = \Phi(x(t_o), t_o, x(t_f), t_f) + \int_{t_o}^{t_f} g(x(t), u(t), t) dt \quad (3.5)$$

subject to the dynamic constraints,

$$\dot{x} = f(x(t), u(t), t) \quad (3.6)$$

the boundary conditions,

$$\phi(x(t_o), t_o, x(t_f), t_f) = 0 \quad (3.7)$$

and the inequality path constraints

$$C(x(t), u(t), t) \leq 0 \quad (3.8)$$

Equations 3.5- 3.8 are collectively referred to as the *continuous Bolza problem* [103, 111, 112]. In Equation 3.5, Φ represents the cost associated with the end points, or the “terminal cost,” and g represents the cost associated with the states and controls along the trajectory, or the “running costs.” In this research, the terminal cost is the final flight time and the running costs are control usage for the duration of the flight. The balance of this section details the specific equations used in the optimal control problem formulation for this dissertation.

3.4.1 UAS Equations of Motion.

The following equations of motion, subject to certain constraints, describe the dynamic model. The UAS airspeed, heading, and pitch angles are three independent controls. The aircraft is defined by six states, \mathbf{x} . The three position states of the UAS are defined along the $x(t)$, $y(t)$, and $z(t)$ axis with respect to the origin in the inertial frame. The next three states are the pitch angle, $\theta(t)$, measured positive up from the inertial xy -plane, the heading angle, $\psi(t)$, measured positive clockwise in the xy -plane the from the inertial x -axis, and the airspeed of the UAS, $V(t)$. The controls are the pitch angle rate in the inertial frame, $\dot{\theta}(t)$, and the heading angle

rate in the inertial frame, $\dot{\psi}(t)$, and acceleration of the UAS, $\dot{V}(t)$. All states and controls are time-dependent and the variable t will be dropped for the balance of the dissertation. The states and controls are bounded by the appropriate limits of the UAS.

The equations of motion are defined by the relationship between the states and the controls. They are derived from the geometry and differential kinematics and are given in Equation 3.9 and consistent with [122]. A geometric definition of the variables is supplied in Figure 11 where the UAS resides at the origin of the coordinate system. The coordinate system is based on an East-North-Up reference frame. East, represented by the state x , positive to the right along the horizontal axis; North, represented by y , positive up along the vertical axis; and Up, represented by z , positive out of the page. Since the relationship of the aircraft to fixed sensors on the ground is being considered, the effect of the wind on the aircraft must also be considered. Therefore, wind velocity components are included in each of the three inertial axes. Aircraft forces and moments are not considered and the UAS performance is adequately represented as a point mass for this research.

$$\dot{x} = V \cos \theta \cos \psi + v_{wind_x} \quad (3.9a)$$

$$\dot{y} = V \cos \theta \sin \psi + v_{wind_y} \quad (3.9b)$$

$$\dot{z} = V \sin \theta + v_{wind_z} \quad (3.9c)$$

$$\dot{\theta} = u_1 \quad (3.9d)$$

$$\dot{\psi} = u_2 \quad (3.9e)$$

$$\dot{V} = u_3 \quad (3.9f)$$

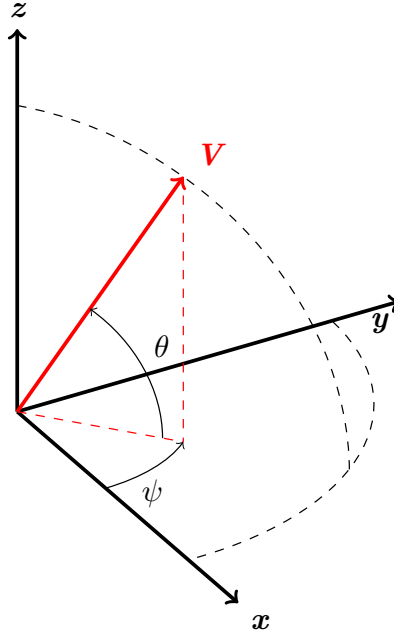


Figure 11. State variables in relation to the inertial frame coordinate axes.

3.4.2 Communications Model.

While the emphasis is in developing an optimal trajectory, it is necessary to include discussion on the communications model. The communications model is the critical link to determining the data rate between the UAS and each sensor; and ultimately to determine the optimal trajectory. To formulate the optimal trajectory, the optimizer requires only a data rate for a given Euclidean distance, d , between the ground sensor and the UAS. The data rate varies as a function of the distance, and optimizing the data rate to meet constraints is the primary dynamic relationship used to determine the optimal trajectory for the UAS. The distance from the sensor to UAS is expressed in terms of the state position variables and the j^{th} sensor's position coordinates, expressed in Equation 3.10. This is a modification to Equation 3.4 where the subscripts are changed to reflect the UAS to sensor distance instead of the

sensor to sensor distance.

$$d_j = \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2}, \quad \forall j = 1 : Q \quad (3.10)$$

Because the communication is intended to collect information stored in the sensor's memory, the parameter of interest is the total quantity of data transferred, measured as in the number of bits. The quantity of data received over the time interval is the integral sum of the data rate, discussed at the end of Section 3.4.2.2. A minimum data requirement is set, thus establishing an integral constraint in the optimal control problem formulation.

The sensors and UAS form traditional client-server architecture. The UAS acts as the server, communicating with all the sensors, acting as clients, when in range. This moving "hub and spoke" model allows multiple ground sensors to connect with the UAS and exchange information simultaneously. The UAS will connect when within the neighborhood of any sensor. Two communications models were considered in this research: the Shannon-Hartley model and an empirically-based emulation.

3.4.2.1 Shannon-Hartley Model.

The physics-based, Shannon-Hartley theorem describes the theoretical communication channel capacity. The channel capacity is the maximum data rate that information can be sent over a certain frequency, f , in the presence of noise, N , with a given bandwidth, B , and transmitter power, P . Equation 3.11 determines the maximum theoretical data rate, or the channel capacity, C , for any given sensor, j , as a function of the distance, d , from the UAS to that particular sensor.

$$C_j = B \log_2 \left(1 + \frac{P}{N \left(\frac{4\pi d_j f}{c} \right)^2} \right) \quad (3.11)$$

The Shannon-Hartley theorem was not chosen for this research because of the uncertainty in the radio parameters and the unnecessary complexity of the model. To effectively implement Shannon-Hartley, the power settings, frequency, bandwidth, and noise must all be known during the UAS flight time. While the communication frequency is set at 2.4GHz, the power and bandwidth are variables the radio software manipulates to maintain a strong signal. Additionally, there are unknown variations in the environmental noise. Determining representative noise, bandwidth and power settings for the communication environment is not a trivial task. A complex communication model is unrealistic for this research and all of the uncertainty involved in identifying the communication environment parameters renders the Shannon-Hartley theorem infeasible.

3.4.2.2 Empirical Communications Model.

A second model was developed which was considerably simpler and based on empirical data from AFRL flight tests. This empirical approach merely emulates the observed behavior of the communications between the AFRL sensors and UAS without modeling the radio hardware. It was observed that the information data rate remains relatively constant over the entire connectivity range. Now, instead of four variables required to determine the communications model, only the maximum connectivity distance is required to have an accurate and representative real-world communications model.

The ground sensors are programmed to transmit messages with a fixed data package size at a constant frequency. When the UAS is within the ground sensor's communication range, it receives data at a constant rate. By maintaining a constant data rate, the only variable becomes whether the sensor and UAS are connected or disconnected. Determining the range where the connection breaks is a tractable

problem. Ground and flight testing yielded sufficient data to determine a nominal communications range.

In 2014, AFRL conducted three flight tests of their UAS and sensor hardware. The data size and transmit frequency were set to a constant data rate of 5 kbit/s. This data rate is far below the available bandwidth to ensure that multiple ground sensors can connect to the UAS simultaneously. A series of varying range tests were conducted to determine the maximum line of sight communication range. The UAS consistently maintained its connection with the ground sensor within a distance of at least 1 km. On several occasions, the UAS continued to receive the constant 5 kbit/s rate for up to two kilometers. However, this was not typical. The maximum connection range was conservatively set to 1 km to ensure that the UAS would always be connected with the ground sensor within that range. AFRL's communication setup features the WiFi 802.11b protocol and behaves in a traditional client and server manner.

The observed communication behavior toggled between two discrete options, connected or disconnected. When connected, a constant rate of 5 kbit/s was observed. For implementation into the optimal control problem, this discrete behavior needed to be transformed into a continuous and differentiable function. A sigmoid function, represented in Equation 3.12 was used because it satisfied the continuous and differentiable requirements, but also allowed for a variation of the stiffness, s , without losing gradient information away from the cutoff value, D_{max} , critical for optimizer search directions. The stiffness feature enabled adjustment of the steepness of the sigmoid and allowed for approximation of a step function. Figure 12 illustrates the empirical model with observed data rates. Two stiffness values were used as examples to illustrate how the model can be modified to mimic the discrete behavior of data

rate past a given distance, in this case 1 km. The data rate was modeled as:

$$C_j = \frac{C_{max}}{1 + e^{-s(D_{max}-d_j)}} \quad (3.12)$$

where d_j is from Equation 3.10, and C_{max} and D_{max} are the maximum communications rate of 5 kbit/s and distance 1 km respectively. As the stiffness is increased, the sigmoid function approaches a step function as shown in Figures 12a and 12b. Figures 12c and 12d show a top view of the sigmoid range and data rates as they emanate from the sensor.

The most accurate representation of the connected and disconnected behavior would require a discontinuous function or a logic statement, both of which are not compatible with the optimal search gradient requirements. Although the sigmoid function is not an exact representation of the observed radio behavior, it is a close approximation. The behavior of the sigmoid based model is also consistent with the work by Kuruvila et al. in Section 2.4.6 and Figure 4. Unfortunately, increasing the stiffness also steepens the gradients in the region surrounding the maximum communication range D_{max} . Increasing stiffness makes the sigmoid function behave more like a step function, as seen in Figure 12b, making it less compatible for gradient search methods. These steep gradients make the optimizer more reliant on the initial guess and establishing feasible search directions more difficult for the optimizer. Without sufficient gradient information, solutions were less likely to converge. Selecting a sigmoid stiffness of 0.01 provided enough gradient information for the optimizer to converge to an optimal solution while closely mimicking the empirical behavior.

The optimal trajectory formulation is agnostic to the particular communication model, so long as the data rate is a function of distance. Although both communication models are able capable of this task, the empirical model was selected for its ease and representation of the observed communication. Based on initial flight test

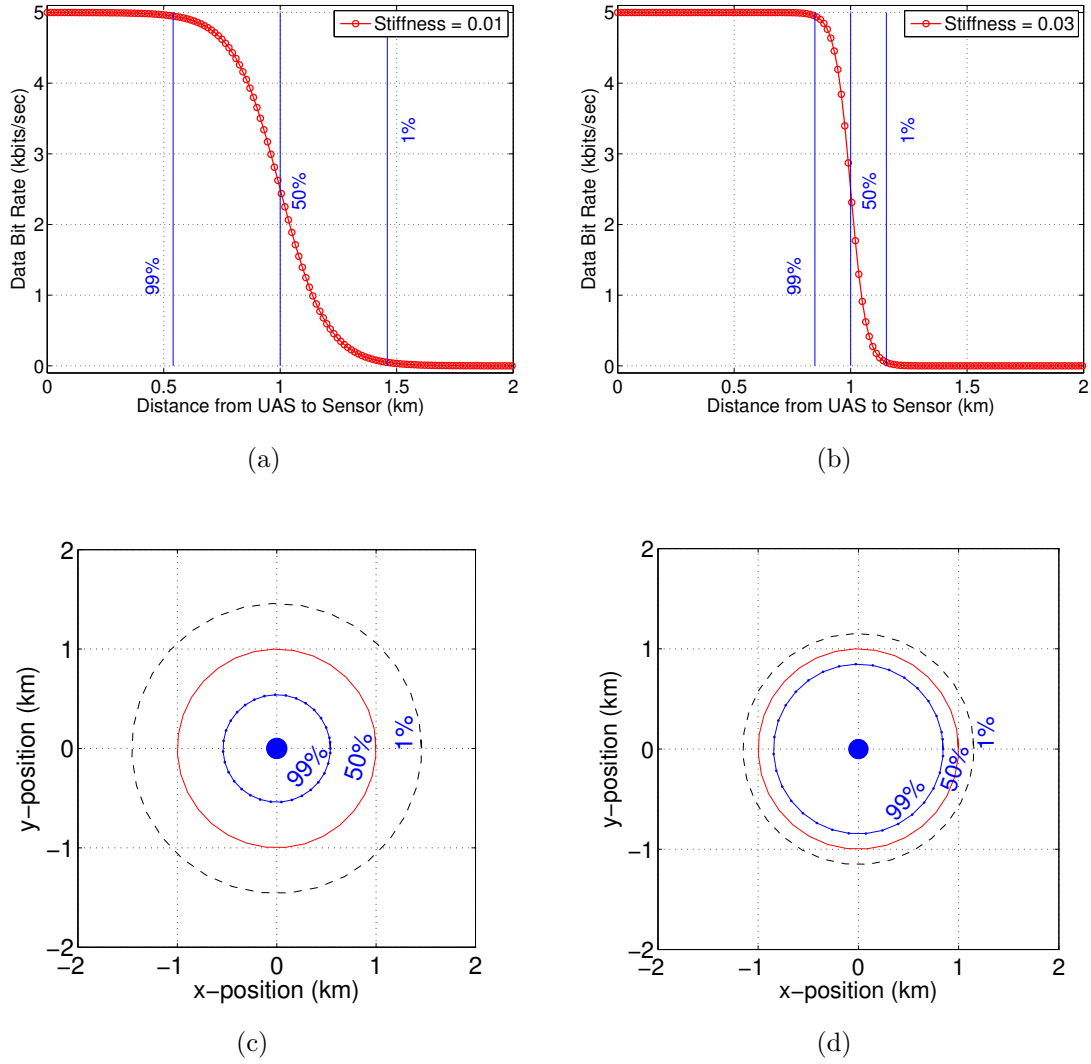


Figure 12. The empirical communications model with a maximum data rate of 5 kbit/s and range of 1km for two stiffness values, (a) $s=0.01$ and (b) $s=0.03$. (c) and (d) represent top views of the communications range and data rate from a sensor for both stiffness values.

results detailed in Chapter V, the nominal values used for research scenarios were set at 5 kbit/sec, a 1 km range, and a stiffness value 0.01.

Finally, the data rate of each sensor is integrated over the time interval of the scenario and the must be equal to or greater than the minimum required data as seen in Equation 3.13. This is an integral path constraint on the optimal control problem

over the entire solution. The minimum data required for each sensor, DP_j , can be uniform for all sensors in a WSN or unique to each sensor.

$$\int_{t_0}^{t_f} C_j dt \geq DP_j \quad \forall j. \quad (3.13)$$

3.4.3 Trajectory Constraints.

To model airspace or terrain limitations, path constraints were incorporated as no-fly zones and minimum altitudes above terrain.

3.4.3.1 No-Fly Zone.

The no-fly zones are represented as path inequality constraints in the optimal control problems, modeled after work by Masternak and Suplisson [105, 106]. Simply stated, the difference between the position states of the k^{th} UAS, \mathbf{x}_k and the center of the no-fly zone, $\tilde{\mathbf{x}}_n$, must be greater than the radius r_n , of the n^{th} no-fly zone. No-fly zones are employed with co-altitude UAS, or to represent obstacles such as mountain peaks or tall buildings.

$$\|\mathbf{x}_k(t) - \tilde{\mathbf{x}}_n\|_2 > r_n \quad \forall n \quad (3.14)$$

3.4.3.2 Intersecting Trajectories.

In lieu of keep-out zones, UAS operating at varying altitude blocks ensure vertical separation from other UAS. Furthermore, by means of problem design, multiple UAS cannot be assigned to the same sensor, so it is highly unlikely that any two trajectories will intersect. This is ensured by hard clustering in the k -means++ and GMM with EM methods.

3.4.3.3 Varying Terrain and Minimum Altitude.

Since the Euclidean distance between the UAS and an assigned sensor is the variable being optimized, the shortest possible distance between the UAS and any sensor is directly above the sensor at the minimum above ground level (AGL) altitude. In three-dimensional scenarios however, if the terrain elevation on which the sensors are placed varies, then the minimum altitude as measured from a reference, sea level for example, must also vary accordingly. To account for a varying minimum altitude, a terrain path constraint was implemented. A terrain model was created based on the elevations of each sensor, where vertices of the terrain are represented by sensor coordinates. Terrain modeling was based on work by Suplisson [105].

The optimization procedure utilizes a gradient search method and requires a continuous and differentiable function representing the terrain at all locations within the bounds of the problem. The continuous function is queried by the nonlinear programming solver at various points to determine if the terrain constraint is violated. So, an interpolation step is required to make the discrete sensor locations into a smooth continuous function. This was accomplished utilizing the MATLAB[®] ‘*scatteredInterpolant*’ function [120]. For the sensor coordinates previously defined in Section 3.2, \mathbf{S} , the ‘*scatteredInterpolant*’ function shown in Equation 3.15.

$$F_{terrain} = scatteredInterpolant(\mathbf{S}) \quad (3.15)$$

Given the scattered, non-uniform, non-gridded locations of the sensors for any given scenario, the ‘*scatteredInterpolant*’ function provided the most efficient method of generating an interpolation function, $F_{terrain}$, to approximate the terrain everywhere. According to Suplisson, a similar interpolation function is “computationally

fast because the interpolating polynomials and coefficients can be determined outside the NLP and then quickly queried in the NLP for path constraint evaluation [105].”

Given the x and y position coordinates of the UAS, $F_{terrain}$ is queried inside the non-linear program solver and returns the interpolated value of the terrain, $z_{terrain}$, at those same coordinates. The difference between the vertical coordinate of the UAS, z , and $z_{terrain}$ must be greater than 100 meters. The 100 meters is added as buffer between the UAS and the terrain. Both steps are shown in Equation 3.16.

$$z_{terrain} = F_{terrain}(x, y) \quad (3.16a)$$

$$z - z_{terrain} > 100 \text{ meters} \quad (3.16b)$$

$z_{terrain}$ plus 100 meters represents the minimum AGL altitude at any location along the terrain and is defined as the minimum path constraint along the z-axis. This minimum path constraint was imposed on both the initial guess and optimization routine calculating the optimal flight trajectory.

3.4.3.4 Boundary Conditions.

The final and initial boundary conditions, except heading, are identical to ensure a closed and smooth closed path (orbit) trajectory over the WSN, where the UAS returns to the initial states. The heading ψ starts at zero and continuously adds and subtracts heading along the trajectory. A full orbit of the WSN starts at 0 radians heading and ends at a final heading of 2π radians. However, the solution software did not recognize them as spatially pointing in the same direction. Without a factor of 2π added to the initial heading, the optimizer attempted to correct back to 0 radians in the final step, hence creating a small, unrealistic heading correction loop as seen in Figure 13. Figure 13b is a notional, smoothed and more realistic optimal trajectory.

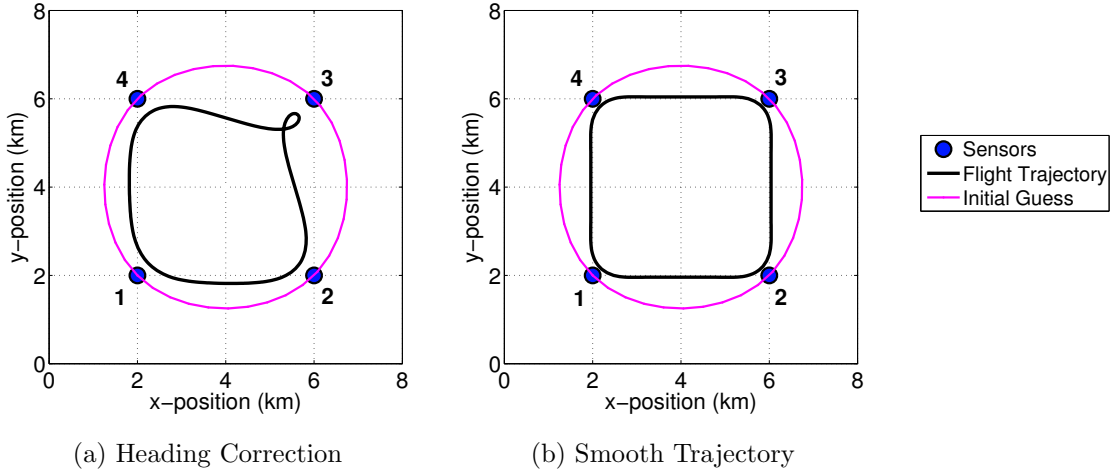


Figure 13. Without the heading correction factor, the heading correction loop is visible in the notional flight trajectory in (a). With the 2π correction, the flight trajectory is smooth.

3.5 Optimal Control Problem Formulation

With the system dynamics, inequality, and integral path constraints, the formal definition of the continuous optimization problem is complete. Time and control weighting factors, α and β respectively, are used to weigh the influence of the individual components of the cost function. The optimal control problem is now formulated to find the optimal control, \mathbf{u}^* , in the set of admissible controls, U , that minimizes the following cost functional

$$\min_{\mathbf{u} \in U} J_k = \alpha (t_{k_f}) + \int_{t_0}^{t_f} \beta_1 \dot{\theta}_k^2 + \beta_2 \dot{\psi}_k^2 + \beta_3 \dot{V}_k^2 dt \quad (3.17)$$

for the k^{th} UAS, where $k = 1, 2, \dots, N$ and N =Total Number of UAS available subject to the dynamic constraints

$$\dot{x} = V \cos \theta \cos \psi + v_{wind_x} \quad (3.18a)$$

$$\dot{y} = V \cos \theta \sin \psi + v_{wind_y} \quad (3.18b)$$

$$\dot{z} = V \sin \theta + v_{wind_z} \quad (3.18c)$$

where the states and controls are

$$\mathbf{x} = [x, y, z, \theta, \psi, V]^T \quad (3.19a)$$

$$\mathbf{u} = [\dot{\theta}, \dot{\psi}, \dot{V}]^T \quad (3.19b)$$

with the boundary conditions,

$$\mathbf{x}(t_0) = [x_0, y_0, z_0, \theta_0, \psi_0, V_0]^T \quad (3.20a)$$

$$\mathbf{x}(t_f) = [x_0, y_0, z_0, \theta_0, \psi_0 + 2\pi, V_0]^T \quad (3.20b)$$

the path constraints,

$$\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \quad (3.21a)$$

$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max} \quad (3.21b)$$

$$\|\mathbf{x}_k(t) - \tilde{\mathbf{x}}_n\|_2 > r_n \quad \forall n \quad (3.21c)$$

$$z - z_{terrain} > 100 \text{ m} \quad (3.21d)$$

and the integral constraints

$$\int_{t_0}^{t_f} C_j dt \geq DP_j \quad \forall j. \quad (3.22)$$

3.6 Gaussian Mixture Model

In order to update the sensor to UAS assignments, sensor clusters are first modeled by a 2D-multivariate GMM. Although sensor coordinates are three dimensional, only the x - and y -coordinates are considered since the magnitude of the z -coordinate has negligible effect on cluster assignment. The GMM is characterized by the 2×2 positive and semi-definite covariance matrix, and the respective coordinate axis means. One Gaussian is required for every existing cluster. Continuing with the same notation started in Section 3.2, \mathbf{S} is the set of all sensors, $\boldsymbol{\mu}_k$ and \mathbf{P}_k are the centroid (means) and covariance of the k^{th} cluster represented a weighted normal distribution. As stated in Section 3.2, the centroid is equivalent to the mean of each cluster. Similarly, the mean of the Gaussian is also the centroid of the cluster generated by the method in this and the next section. For all the sensors, the probability density, p , in equation 3.23, defines a GMM as the sum of multiple weighted Gaussians,

$$p_{\mathbf{S}} = \sum_{k=1}^N w_k \mathcal{N}(\mathbf{S} | \boldsymbol{\mu}_k, \mathbf{P}_k) \quad (3.23)$$

where $w_k \in \{0, 1\} \forall k$.

3.7 Expectation Maximization

The EM Algorithm 2 updates the GMM as high interest sensors are identified. High interest sensors receive increased weighting in the EM process. This effectively shifts the Gaussian mean towards the higher weighted sensors, and along with the covariance, reshapes the Gaussian distribution, capturing a new set of sensors and redefining the cluster. The weighting of the Gaussian, w_k , in the EM process is not the same as the weighting, w_h , applied to high interest sensors. Individual Gaussian weights, w_k , are updated in Algorithm 2 at step 7, and contribute to determining

the new means and covariances of the updated Gaussians during the Maximization steps. In contrast, high interest sensor weights, w_h , are input prior to beginning Algorithm 2 and do not change during execution of the algorithm. Specifically, the individual sensor weights are incrementally increased between each run Algorithm 2 to produce different clustering. Incrementation of w_h is illustrated in Chapter 4, Figures 35 and 29. For each sensor weighting incrementation, the EM algorithm is repeated. Treating an unknown quantity of observations and components is often cited in the GMM and EM research. However, in this dissertation, the two processes are eased since the exact number and location of components (centroids) and observations (sensors) is known *a priori* and do not necessarily change through the given scenarios. To begin the process of sensor to UAS reassignment, the GMM utilizes the *k-means++* centroids from Algorithm 1 as the initial means of each Gaussian.

3.8 Reference Scenarios

Two reference scenarios are used to better help the reader compare similar scenarios under various conditions. The first employs one UAS surveying six sensors on two elevations and the second employs ten co-planar sensors with two UAS. The sensor locations in the six sensor scenario were deliberately chosen because their placement lend themselves to more intuitive solutions, easily interpreted by the reader. The second scenario is comprised of ten randomly scattered sensors generated by MATLAB's[®] *rand* function designed to represent a random fictitious WSN. Other ancillary test cases are used to highlight particular results as needed, but may not persist throughout the dissertation.

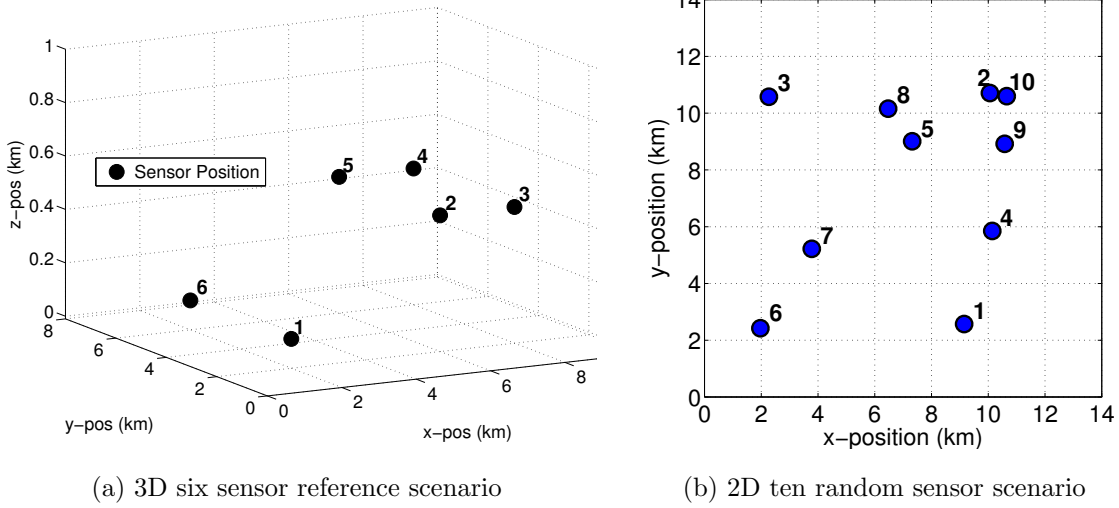


Figure 14. Reference Scenarios.

3.9 Summary

This chapter provided an overview of the methodologies used in this dissertation. An overview of the end to end process was depicted in a simple flow chart. Two sensor to UAS assignment approaches were presented, as well as three initial guess methods, aircraft and communication models, and an optimal trajectory problem and constraints were posed. Finally, two reference scenarios were presented. The next chapter presents the details required to employ and solve the methodologies presented utilizing the reference scenarios and other test cases.

Algorithm 2 Expectation Maximization [97, 98, 119]

Inputs: $Q \times 2$ Sensor coordinates $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_j, \dots, \mathbf{s}_Q\}$, $N \times 2$ Initial cluster means (centroids) $\mathbf{C} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k, \dots, \boldsymbol{\mu}_N\}$, $H \times 2$ High activity sensor coordinates $\mathbf{S}^* = \{s_1^*, \dots, s_j^*, \dots, s_H^*\}$, and weight factor for active sensors, w_h .

Output: Updated cluster means (new centroids) $\bar{\mathbf{C}} = \{\bar{\boldsymbol{\mu}}_1, \dots, \bar{\boldsymbol{\mu}}_k, \dots, \bar{\boldsymbol{\mu}}_N\}$ and Updated Sensor Clusters $\bar{\mathbf{S}}_j^c = \{\bar{s}_1^1, \dots, \bar{s}_j^k, \dots, \bar{s}_Q^N\}$.

- 1: Concatenate \mathbf{S} with w_h multiples of \mathbf{S}^* : $\bar{\mathbf{S}} = [\mathbf{S}, (w_h - 1) \times \mathbf{S}^*]^T$ with length \bar{Q}
- 2: Determine x - and y -coordinate variances for k clusters: $\sigma_j^2 = \text{var}(\bar{S}_1, \dots, \bar{S}_j)$
- 3: Set initial cluster covariances equal: $\mathbf{P}_k = \text{diag}[\sigma_1^2, \dots, \sigma_N^2]$
- 4: Set initial cluster weights equal $w_k = \frac{1}{N}$
- 5: Form initial Gaussian mixture with input means: $p_S^M = \sum_{k=1}^N w_k \mathcal{N}(\mathbf{S} | \boldsymbol{\mu}_k, \mathbf{P}_k)$

Expectation Step

- 6: Compute $\bar{Q} \times N$ test membership weight matrix for each sensor in each cluster:

$$\mathbf{M}_{ij} = w_k \mathcal{N}(\mathbf{S} | \boldsymbol{\mu}_k, \mathbf{P}_k) / \sum_{k=1}^N w_k \mathcal{N}(\mathbf{S} | \boldsymbol{\mu}_k, \mathbf{P}_k)$$

Maximization Step

- 7: Determine new weights with $1 \times N$ vector \mathbf{m} : $\mathbf{m}_k = \sum \bar{Q} \mathbf{M}_{ik} w^{new} = \frac{1}{\bar{Q}} \mathbf{m}$
 - 8: Determine new means and covariances for $k = \{1, \dots, N\}$:

$$\boldsymbol{\mu}_k^{new} = \frac{1}{m_k} \sum \bar{Q} \mathbf{M}_{ik} \mathbf{S}_i$$

$$\mathbf{P}_k^{new} = \frac{1}{m_k} \sum \bar{Q} (\mathbf{M}_{ik} \mathbf{S}_i - \boldsymbol{\mu}_k^{new}) (\mathbf{M}_{ik} \mathbf{S}_i - \boldsymbol{\mu}_k^{new})^T$$
 - 9: Form updated mixture: $p_{S_{new}}^M = \sum_{k=1}^N w_k^{new} \mathcal{N}(\mathbf{S} | \boldsymbol{\mu}_k^{new}, \mathbf{P}_k^{new})$
 - 10: Calculate negative log likelihood: $q_{new} = -\ln(p_{S_{new}}^M)$
 - 11: Iterate, returning to step 6 until: $(q_{new} - q_{old}) < tolerance$
 - 12: When tolerance met: $\boldsymbol{\mu}^{new} = \bar{\boldsymbol{\mu}}$ and $\mathbf{S}_{new} = \bar{\mathbf{S}}$
-

IV. Results

In this chapter, the methodologies detailed in Chapter III are implemented and results presented by means of examples. As a reminder to the reader, the topics in this chapter are generally ordered in the manner they are used to solve the problem; first, sensors are clustered, the initial guess formulated, and clusters are assigned to UAS; second, optimal trajectory is formulated; and finally, based on sensors identified as high interest, a re-clustering and re-assignment procedure is completed. Problem setup information, specific values, and software utilized are presented to provide a complete picture of the solutions. Simulation results use a real-world example test case as well. Initial results were presented at the AIAA SciTech Conferences in January 2014 and 2015 and the corresponding papers are provided as Appendices A, B, and C.

4.1 Problem Setup, Settings, Values, and Software

The following sections specify the setup of the problem, to include UAS performance limits, selected software settings, and all parameters used for the solutions presented. Also, a short discussion on the solution approach is included.

4.1.1 UAS State and Control Limits.

UAS performance limits are derived from various sources and consistent with nominal DoD Group 1 UAS and commercial small UAS in use today [123]¹². State and control limits were more conservative than the actual performance limits of the nominal UAS. This enabled a solution space for a wider array of possible UAS. Addi-

¹Reference to the MartinUAV company SuperBAT and BAT 4 UAS for example, <http://martinuav.com/products-s-bat/>

²Email correspondence with Dr Dave Gross, ICE-T Principle Investigator, Wright State Research Institute and Dr Derek Kingston, AFRL

tionally, the limits listed in Table 2 are representative of aircraft used for flight testing later in Chapter V.

Table 2. UAS State and Control Limits.

| | | Minimum | Maximum | Units |
|----------|---------------------------------|-------------------------|-------------------------|-----------|
| States | x-location | $\min\{S_x, x\} - 2000$ | $\max\{S_x, x\} + 2000$ | m |
| | y-location | $\min\{S_y, y\} - 2000$ | $\max\{S_y, y\} + 2000$ | m |
| | z-location | $\min\{S_z, z\} + 100$ | $\max\{S_z, z\} + 100$ | m |
| | Heading angle θ | -180 | 180 | deg |
| | Pitch angle ψ | -15 | 15 | deg |
| | Velocity \mathbf{V} | 10 | 30 | m/sec |
| Controls | Heading rate $\dot{\theta}$ | -11.5 | 11.5 | deg/sec |
| | Pitch rate $\dot{\psi}$ | -6 | -6 | deg/sec |
| | Acceleration $\dot{\mathbf{V}}$ | -2 | 2 | m/sec/sec |

4.1.2 Software.

The solutions presented in this chapter and in Chapter V utilized several software packages. MATLAB[®] by MathWorks was the primary software used to solve the assignment, initial guess, and optimal trajectory problems. The MATLAB[®]-based, General Purpose Pseudospectral Optimal Control Software-II (GPOPS-II) developed by Rao and Patterson was used to solve for the optimal trajectories [124] and discussed in detail in Section 4.4.1. Also, the Air Vehicle Test Analysis and Simulation Multi-Agent Simulation Environment (AMASE) suite from AFRL is used to simulate the optimal trajectory and Excel by Microsoft is used to help prepare flight test data for processing. AMASE is discussed in more detail in section 4.6. The results presented were performed using MATLAB[®] 2014a on a mid-2012 MacBook Pro[®] with a 2.9 GHz Intel Core i7[®] processor, 16 GB of 1600 MHz DDR3 memory, and a 120GB solid state drive.

4.2 Sensor Clustering

The clustering and assignment methodology provided a quick and logical solution. Sensors were appropriately allocated to centroids and subsequently to respective UAS. For the ten sensor reference scenario, the randomly placed sensors were divided into two clusters (since two UAS were made available). Figure 15 illustrated the clustering of ten sensors into two clusters. The k -means++ clustering (Algorithm 1) was implemented via a MATLAB[®] routine.

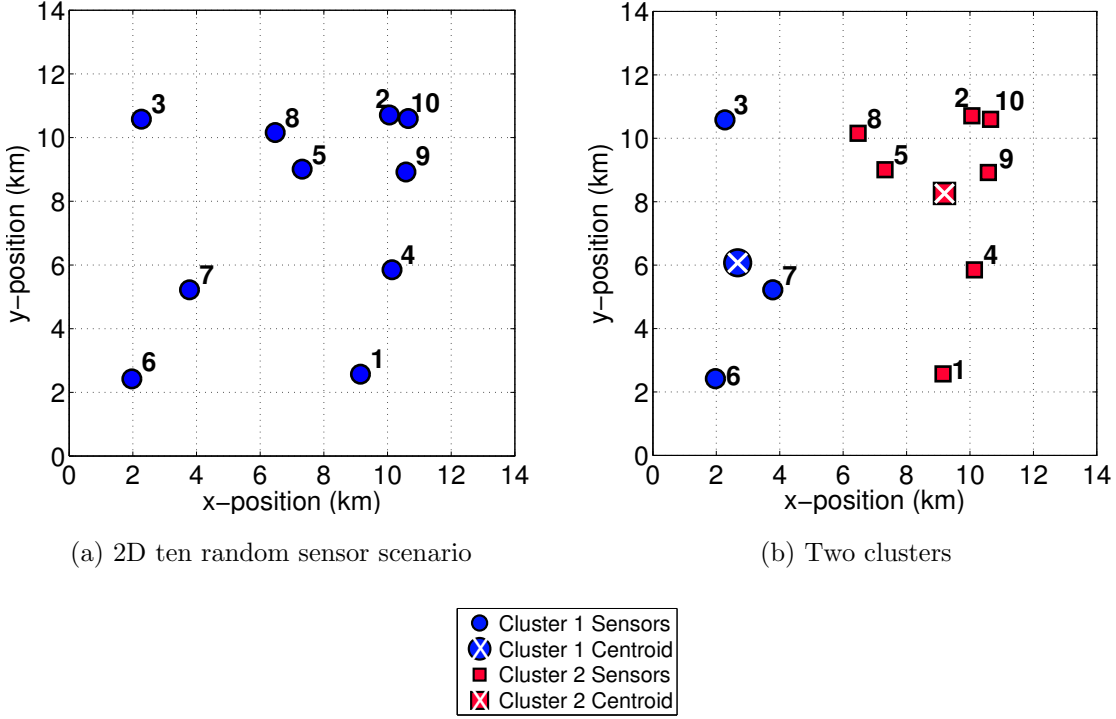


Figure 15. 2D ten random sensor scenario (a) and split into two clusters (b) after k -means++.

4.3 Initial Guess of the Optimal Trajectory

Three initial guess methods were tested in the Six Sensor reference scenario. Table 3 lists the various results for the different initial guesses. The Six Sensor reference

scenario optimal trajectory is also discussed later in the chapter. For all the three test cases, the cost function weights were fixed. The GPOPS-II scaling method was set to *automatic-guess* to prevent GPOPS-II from rescaling the problem and updating the initial guess. The results presented here were consistently repeatable and all resulted in nearly identical optimal flight paths collecting the minimum required data. All three approaches to the initial guess returned the same optimal trajectory at largely the same mesh error. However, the Spline fit TSP initial guess resulted in fewer iterations to solve and a faster computation time, as listed in Table 3. The computation time is subject to the computer hardware used, but since the same computer was used for all three test cases, it proved to be a valuable factor to consider.

Table 3. Influence of various initial guesses as applied to the Six Sensor reference scenario with cost function weights fixed at $\alpha=0.01$ and $\beta=0.1, 1$, and 0.001 respectively.

| Initial Guess Method | Two Point | TSP | Spline fit TSP |
|----------------------|--------------|--------------|----------------|
| Cost | 6.462 | 6.462 | 6.462 |
| Flight Time | 644 sec | 644 sec | 644 sec |
| Distance Flown | 19 km | 19 km | 19 km |
| NLP Time | 23.1 sec | 22.8 sec | 32.1 sec |
| Collocation Points | 229 | 232 | 247 |
| Mesh Error | $8.38e^{-7}$ | $8.43e^{-7}$ | $8.31e^{-7}$ |
| Iterations | 7 | 5 | 5 |
| Computation Time | 213.7 sec | 142.2 sec | 136.8 sec |

Given the ten-sensor reference scenario, the left hand cluster in Figure 16a effectively has one solution to connect the three sensors, but the right hand cluster has seven factorial, or 5,040, combinations. The initial TSP solution helps to narrow that initial search space. If the ten-sensor reference scenario employed a single UAS, the various combinations would number 3,628,800. The spline fit TSP initial guess

results in a continuous and smooth input. Furthermore, the TSP initial guess (I.G.) tells the optimal trajectory routine where all the sensors are located; this is consistent with the problem assumptions in Chapter I. Later in section 4.4.7, two initial guess

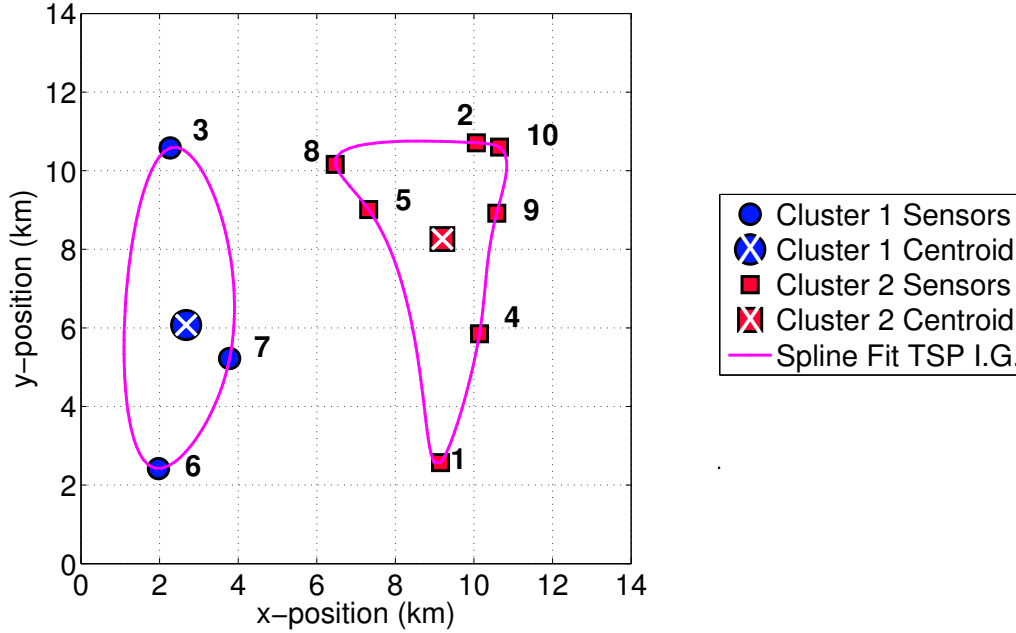


Figure 16. Two clusters of sensors with TSP initial guess.

methods are tested in a ten sensor scenario as well.

4.4 Optimal Trajectory

Optimal trajectories are found for 2D and 3D scenarios. Simpler test cases are used as baseline checks of the optimization routine, where the solution is easily interpreted. Later, more complex scenarios are evaluated. The problem was solved incrementally to better understand the effects of subsequent layers of complexity. Initially, only a single sensor is considered and only the minimum required data was changed. Subsequent scenarios varied other parameters such as various minimum

data required, sensor elevations, path constraints, and available UAS. Finally, the results of a multi-UAS, ten-sensor scenario are presented.

All solutions are calculated for the 3D case and utilize the equations of motion from section 3.4.1, except for the first Single Sensor Single UAS scenario. 3D plots of solutions are only presented for scenarios where the sensors are at various elevations. Otherwise, solutions are presented in a top down, 2D view. For 2D solutions, the UAS flew at the minimum allowable altitude, thereby decreasing the distance to each sensor, and maximizing the available data rates. Calculating a purely 2D solution will produce a solution representative of a ground vehicle and not account for the vertical distance from the sensor to the UAS, a critical component in determining the optimal trajectory.

All solutions presented met specified optimality criteria and error tolerances. The empirical communications model in section 3.4.2.2 and in Figure 12, is represented by concentric rings of 99%, 50%, and 1% of maximum data rate from the sensor and are displayed in the top down 2D flight trajectory views. This is done to aide the reader and indicates when UAS is in range of a particular sensor.

4.4.1 Optimization Software - GPOPS II.

The MATLAB[®]-based, open source, General Purpose Pseudospectral Optimal Control Software-II (GPOPS-II) is the program used to solve for the optimal trajectories. GPOPS-II directly solves the continuous time optimal control problem [125]. GPOPS transcribes a continuous-time optimal control problem into a discrete finite-dimensional static optimization problem that is solved as a Non-Linear Programming (NLP) problem. GPOPS-II is implemented using the *hp*-adaptive version of the Radau pseudospectral method. This particular method is an orthogonal collocation Gaussian quadrature implicit integration method in which the collocation is

performed at the Legendre-Gauss-Radau quadrature points [124]. An adaptive mesh refinement method is implemented that determines the number of mesh intervals and the degree of the approximating polynomial for each mesh interval in order to satisfy the specified error tolerance [124]. GPOPS-II can be set to prioritize finding the degree of the approximating polynomial before refining the mesh interval or vice versa. GPOPS-II’s strength is the ability to divide and add intervals as needed. This key capability, makes it more desirable than generic pseudospectral solvers [106].

4.4.2 GPOPS-II Settings.

For all the test cases, GPOPS-II is set to use the IPOPT NLP solver. The derivative approximation used is “sparseCD” and second derivatives are provided. The mesh method is set to “hp-LiuRao,” where GPOPS-II increases the degree of the approximating polynomial up to a maximum allowable degree prior to dividing the interval into sub-intervals [126]. The error tolerance is set to 10^{-3} . “RPMintegration” is enabled and manual scaling is performed. The significant GPOPS-II settings are listed in Table 4.

4.4.3 Cost Function Weights.

The cost function weights were determined heuristically. Upon examination of the output, cost function weights are adjusted up or down to allow the aircraft to take advantage of the available performance envelope. The user can specify the influence of the time, acceleration, pitch and heading rate terms simply by increasing or decreasing the weighting values. For example, if the user desires to lower flight time, the weighting on time is increased. In order to satisfy all the optimality criteria, setting any weights to zero is not recommended. Including all the cost function terms helps to ensure control inputs are smooth and continuous, behaving as close to actual

Table 4. GPOPS-II Settings.

| | |
|--|-------------------------------------|
| mesh.method | hp-LiuRao |
| mesh.tolerance | 10^{-3} |
| mesh.maxiterations | 2 |
| setup.nlp.solver | IPOPT (v3.11.0) |
| setup.nlp.ipoptoptions.linear_solver | ma57 |
| setup.nlp.ipoptoptions.tolerance | 10^{-6} |
| setup.nlp.ipoptoptions.maxiterations | 2000 |
| setup.derivatives.supplier | sparseCD (sparse Center Difference) |
| setup.derivatives.derivativelevel | second |
| setup.method | RPM-integration |
| min num of collocation points per interval | 3 |
| max num of collocation points per interval | 10 |
| setup.scales.method | automatic-guess |

aircraft flight controls as possible. Otherwise the effect of any particular control input is left unchecked and may behave erratically. The α weighting allows the user to tune down the most dominant term, time, while the β weights help to smooth out control inputs and ultimately the optimal trajectory.

The cost function weights remain subjective and largely up to the interpretation of the user. Each term in the cost function changes in magnitude with various iterations of the same problem. This is especially true for various scenarios. Normalization and development of an expression to solve for the weighting on each iteration or each problem remains problematic. The best method is simply to tune the weights for various problems. However, after examining hundreds of various problems, and

iterations of those problems, the most widely applicable settings are $\alpha = 0.01$, $\beta_1 = 0.1$, $\beta_2 = 1$ and $\beta_3 = .001$.

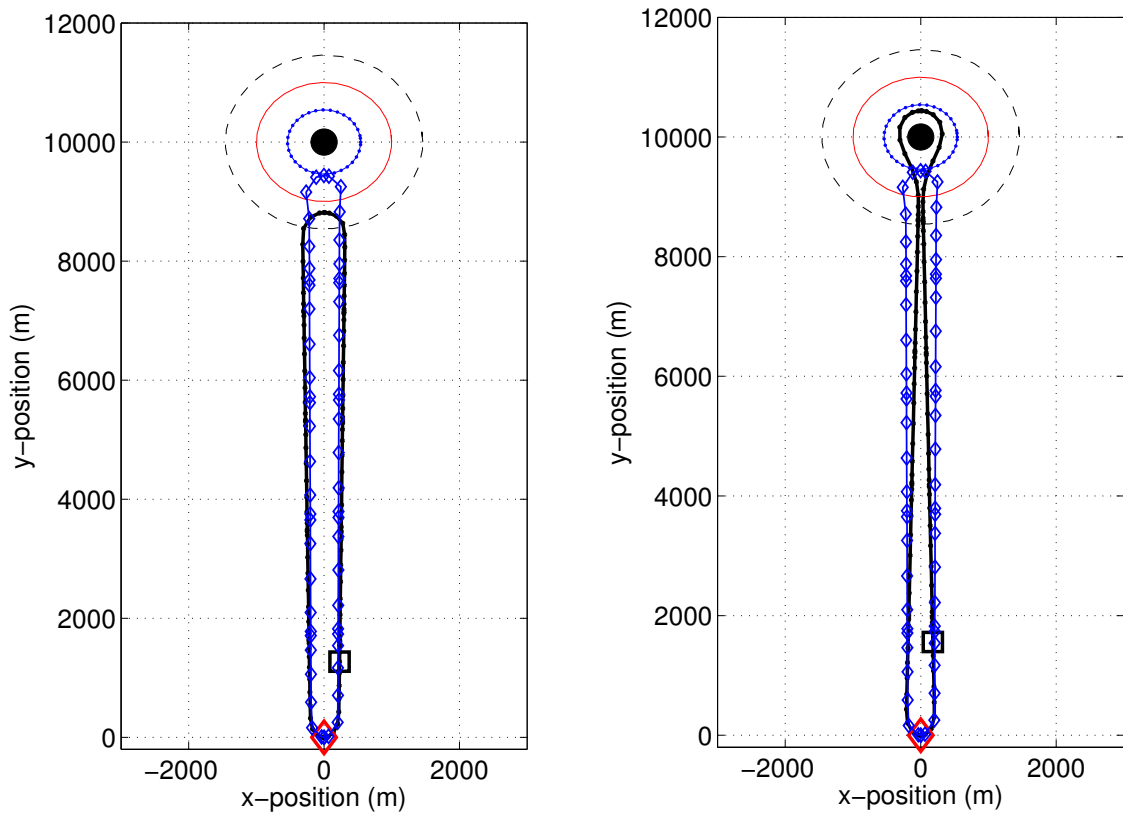
4.4.4 Single Sensor Coplanar with One UAS.

The single sensor scenario has a more intuitive and easily interpreted solution. For this scenario, the UAS departs and returns in the same heading from a fictitious airfield. The take-off and landing phases are assumed negligible and the problem is solved in two-dimensions. Figure 17 illustrates the optimal flight trajectory from the airfield, collecting 10 and 600 kbit of data respectively from a single sensor, and returning to the airfield. The resulting data collection rates are presented in Figure 18, and the required pitch, heading, and velocity in Figure 19. The data rates and controls are provided for the reader's understanding and will not always be presented.

These examples illustrate the change in the optimal trajectory solution as a result of increased data demand. The first test case resulted in a flight time of 613 seconds compared to 735 seconds for the second, which also took a longer flight trajectory around the sensor in Figure 17b.

The data rates are also reflected in Figure 18 and verifies the integral constraint. As expected the data rate increases as the UAS approaches the sensor, and decreases as it flies further away. In Figures 18a and 18b, total area under the plot is equal to the minimum data required of 10 and 600 kbit respectively.

The UAS pitch, heading, and velocity states shown in Figure 19 also verify the flight trajectory plots. After takeoff, the UAS accelerates to the maximum velocity for the duration of the scenario. However in the 600 kbit solution in Figure 19b, the optimizer reduced the airspeed velocity to ensure it collected the required data. The velocity change is clearly seen at the same time of the most significant heading change, indicating the vehicle slowed down at the sensor to take advantage of the maximum



(a) Optimal trajectory for UAS collecting 10 kbit (b) Optimal trajectory for UAS collecting 600 kbit

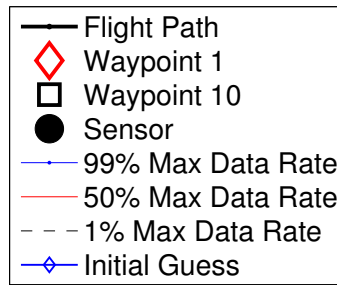
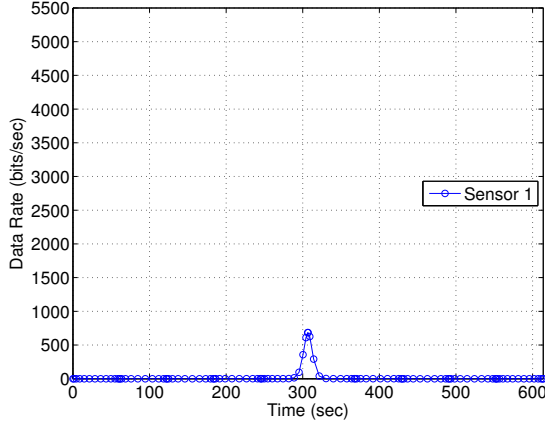
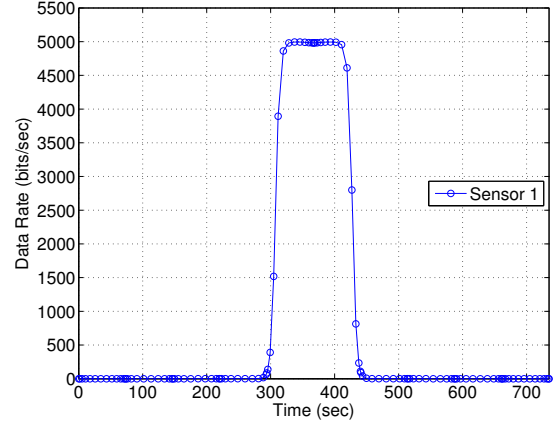


Figure 17. (a) and (b) show the optimal UAS trajectories to collect 10 and 600 kbit of data from a single sensor respectively.

data rates. This simple scenario verifies the basic functionality of the optimization routine.

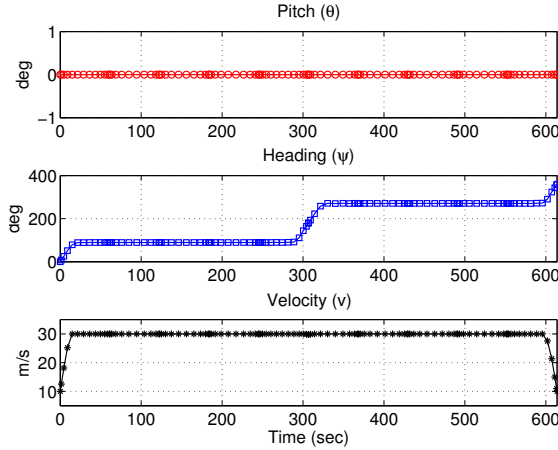


(a) Data rate required to collect 10 kbit

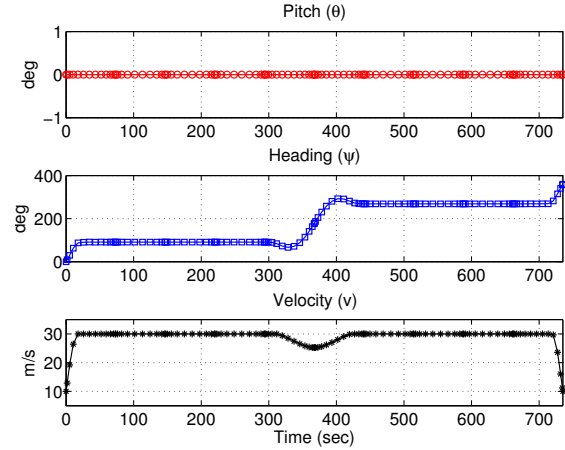


(b) Data rate required to collect 600 kbit

Figure 18. (a) and (b) show the data rates resulting in 10 and 600 kbit of data from a single sensor from trajectories in Figure 17 respectively.



(a) UAS states to collect 10 kbit



(b) UAS states to collect 600 kbit

Figure 19. (a) and (b) show the pitch, heading and velocity states from the 10 and 600 kbit data collections in Figure 17 respectively.

4.4.5 Six Sensors on Two Planes with One UAS.

Figure 20 illustrates the Six Sensor reference scenario. Sensors are placed on two planes, two at a 100 m elevation and four at a 500 m elevation respectively. As in the single sensor test case, the solution to this reference scenario is intuitive and

highlights the three-dimensional capability. In this scenario, the UAS is required to collect 300 kbit of data from each sensor.

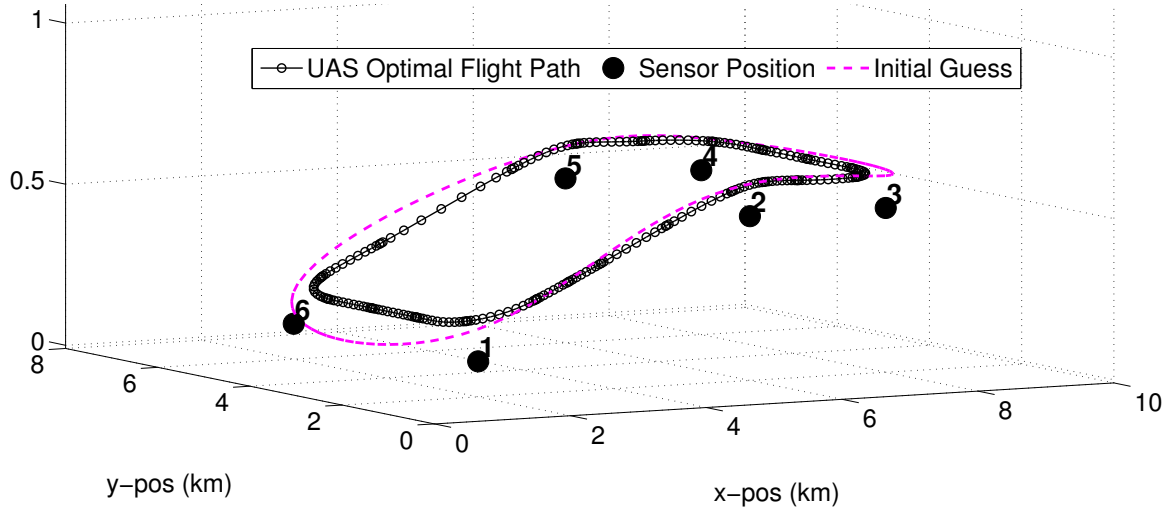
In the Six Sensor scenario, the UAS successfully collected 300 kbit of data without violating any constraints. The trajectory (path) constraint is illustrated in Figure 20b with the sensors located at the vertices. The trajectory constraint is necessary, especially in the three-dimensional case, to ensure the UAS trajectory does not fly below the elevated sensors. The trajectory constraint acts like terrain, requiring the optimization to remain above it at all times.

Control usage was more pronounced in this scenario as pitch varied as well, illustrated in Figure 20c. The optimizer employed a slower velocity over sensors #3, #4, #6, and #1 to ensure the minimum data was collected.

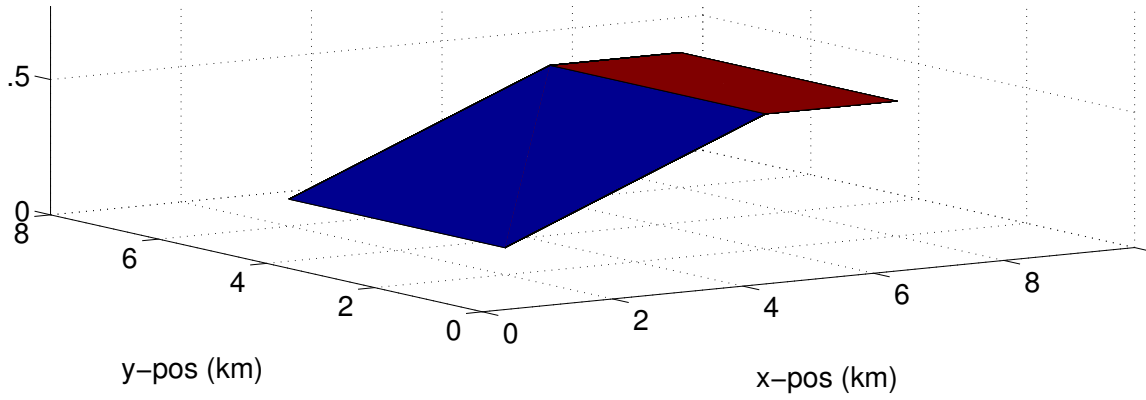
4.4.6 Sixteen Gridded Sensors with One UAS.

In this scenario a gridded WSN of sixteen sensors was tested with one UAS. The initial lay out is shown in Figure 21. The UAS was tasked with collecting 10 kbit from each sensor. The optimal trajectory is illustrated in Figure 22a. After the initial orbits, the square shaped sensors report higher activity within their sensing range. The initial flight time was 799 seconds for the UAS to collect the required 10 kbit from each sensor.

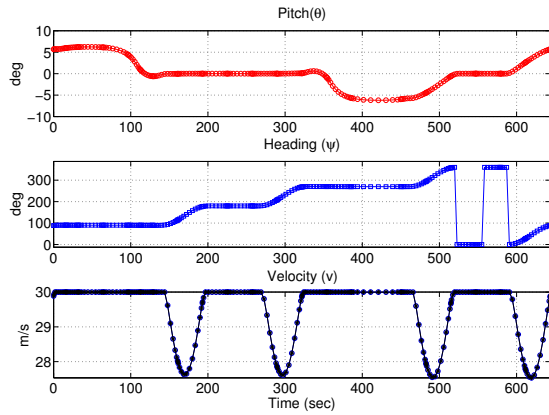
The amount of data required from the high activity sensors is arbitrarily selected at 1 Mbit, 100 times the amount of the other sensors. Once the high activity sensors were identified, a new optimal trajectory was calculated. The new optimal path ensured the UAS concentrated more on the high interest sensors by reducing velocity and increasing distance within thier communications range. The updated trajectory is shown in Figure 23a. The new flight path shows the UAS inside the highest data rate region of the four sensors of interest. In addition to the flight path, the vehicle



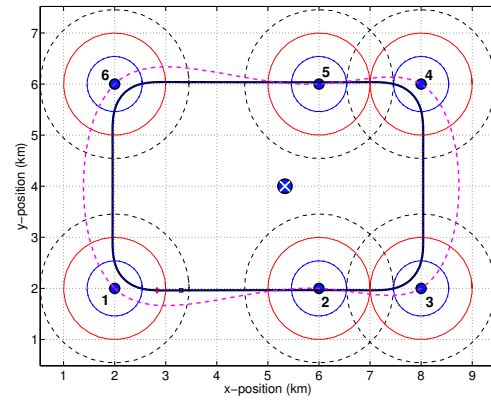
(a) 3D flight path - isometric view



(b) Path Constraint



(c) Pitch, heading, and velocity



(d) 3D flight path - top view

Figure 20. Three-dimensional, optimal UAS trajectory for a six sensor test case where four sensors are elevated 500 meters. For (d) refer to the legend in Figure 20a.

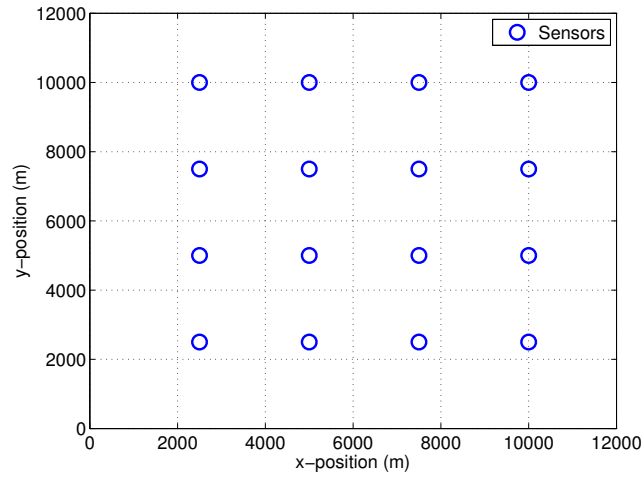


Figure 21. Sixteen sensor gridded layout example for single UAS scenario.

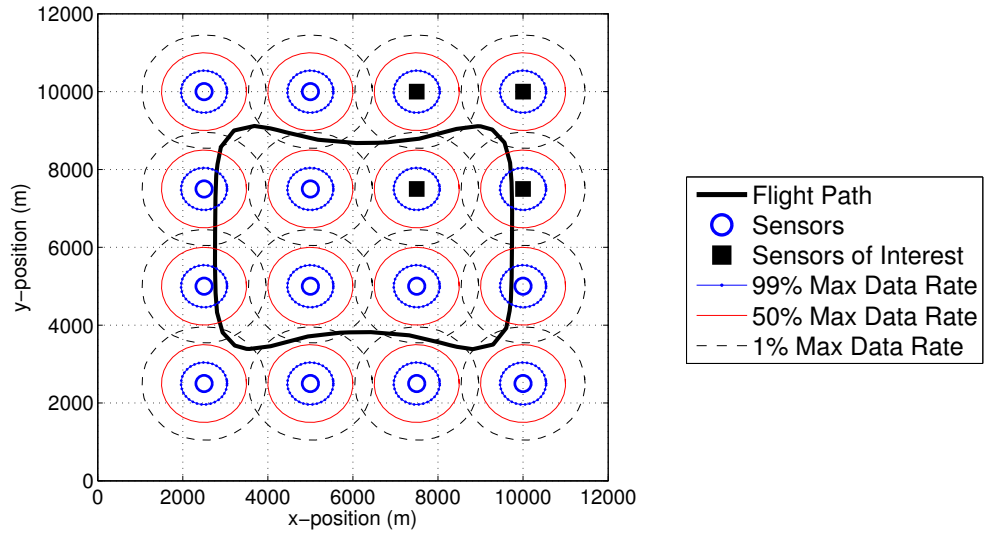


Figure 22. The optimal flight trajectory for the sixteen sensor gridded layout for one UAS. The square sensors identify sensors with high activity after the initial orbit (for use in the subsequent figure).

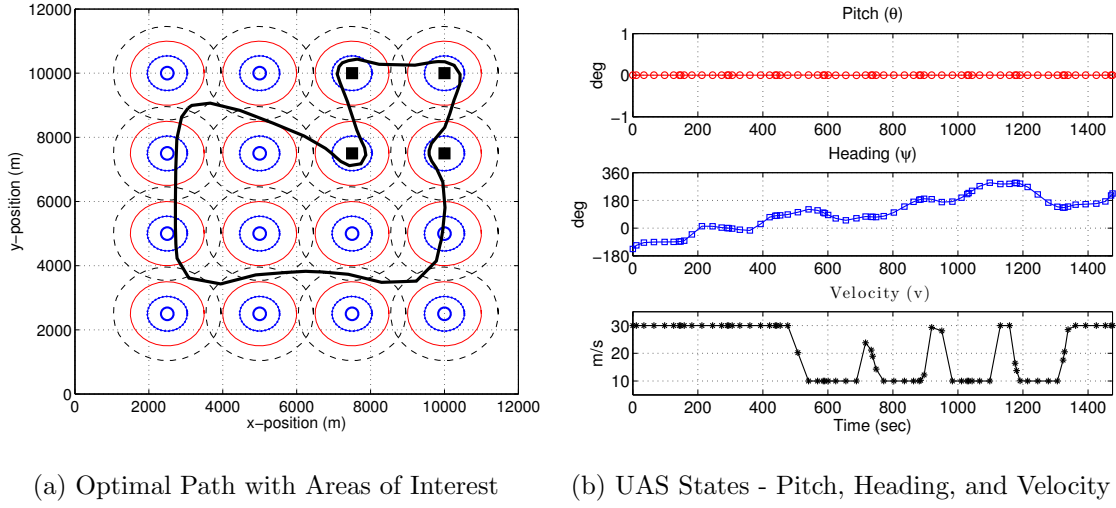
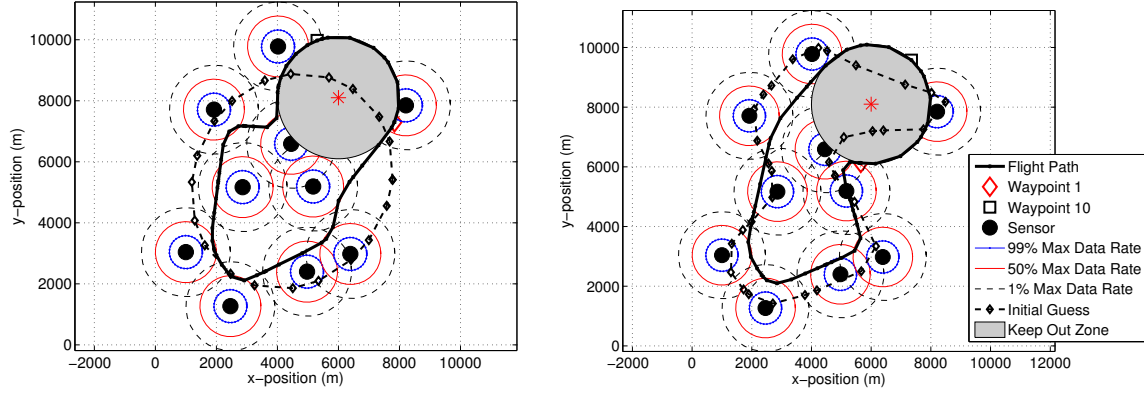


Figure 23. Re-planned optimal paths for two UAS given updated sensor areas of high activity.

states are also worth examining. Figure 23b illustrates the change in velocity state from its maximum of 30 m/s to 10 m/s over the four sensors of interest. Slowing down in the regions of high data rates allows the UAS to optimally collect the minimum data required. After collecting the minimum data from one sensor it noticeably accelerates to the next sensor before slowing again. This is due in large part to α and β settings, where total flight time and heading rate are penalized more than velocity. The total flight time for the updated trajectory was 1,474 seconds. The UAS spent approximately 550 seconds on the 12 low activity sensors and 921 seconds on the four high activity sensors. In other words, the UAS spent approximately 134 seconds with each high interest sensor and approximately 77 seconds on each of the remaining sensors. The optimizer confirmed the four high activity sensors uploaded 1 Mbit each and the remaining sensors upload the minimum 10 kbit. This demonstrates the optimization method capability to persist over the region of higher interest while maintaining contact with the sensors in the region of less interest.



(a) Flight trajectory with circular initial guess (b) Flight trajectory with TSP initial guess

Figure 24. Optimal UAS trajectory for 10 random sensors with no-fly zone and initial guess.

4.4.7 Ten Sensors with One UAS and No-Fly Zone.

Figure 24 illustrates a separate ten random sensor test case. In this scenario, sensors are placed around the no-fly zone that may represent a mountain or other obstruction. The UAS was tasked with collecting 100 kbit of data from each sensor. For this scenario two different initial guesses were used to determine their impact on the optimal trajectory.

Despite two different initial guesses, optimal solutions were found for both inputs. Figure 24a began with a more circular initial guess, generated without the no-fly zone constraint, resulting in an optimal flight time of 751 seconds. Figure 24b shows the TSP initial guess, also generated without the no-fly zone constraint. This resulted in an optimal flight time of 776 seconds, 25 seconds longer than the optimal trajectory resulting from the circular initial guess, or a 3% difference.

In addition to comparing the final time of the two optimal flight trajectories, the data collected from each sensor is illustrated in Figure 25. The black bars are data collected as result of the optimal trajectory generated from the circular initial guess while the orange bars are data collected from the optimal trajectory as a result of the

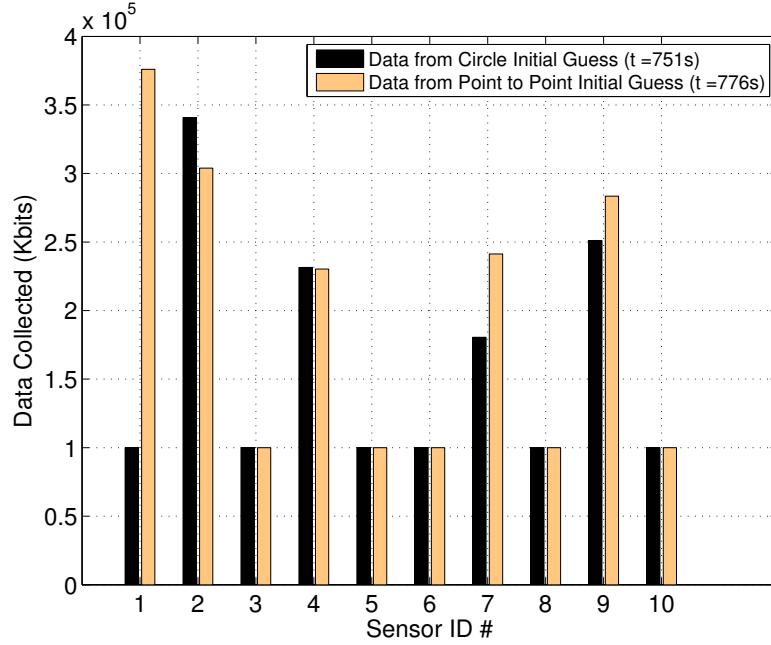


Figure 25. Data collected from the ten sensor, no-fly zone scenarios given two initial different guesses.

TSP initial guess. The circular initial guess resulted in an optimal trajectory that collected 100 kbit from six sensors. From the remaining four sensors, 340, 231, 180, and 251 kbit were collected respectively. The total data collected as a result of the circular initial guess was 1604 kbit. Using the TSP initial guess resulted in an optimal trajectory that collected a 100 kbit from five sensors. The UAS also collected 376, 303, 230, 241, and 283 kbit from the remaining five sensors respectively. The total data collected using the point to point initial guess was 1935 kbit, or 331 kbit more than the optimal trajectory that resulted from the circular initial guess.

Only the initial guess changed between the two problems, and clearly had an influence on the optimal trajectory. The two results represent local optimal solutions and neither may be the global solution. Factors such as cost functional weights, total data required, maximum communication range and rates, communications model stiffness, and UAS performance parameters are all subjective and easily influence the outcome.

These values depend largely on the hardware limitations, mission parameters, and user desires. Tighter tolerances on the total data collected for each could be imposed, however this over-constrains the problem and may prevent convergence. Overall, the ability to determine an optimal trajectory around an exclusion zone demonstrates robustness to increasing problem demands while communicating with all the sensors.

4.4.8 Ten Sensors with Two UAS.

Figure 26 illustrates the optimal trajectory solutions for multiple UAS in the ten-sensor reference scenario. The UAS were tasked to collect 300 kb of data from each sensor. This is the baseline set up for the re-assignment scenario in section 4.5. The controls and data rates in Figure 27 confirm the UAS performed within the established bounds and are consistent with earlier, simpler examples. The UAS fly their assigned trajectory, collect the necessary data, and identify sensors of interest based on the data reported.

4.5 Sensor Reassignment for Ten Sensor Two UAS Scenario

After the initial orbits of the WSN by the two UAS in the ten-sensor reference scenario, sensors #9 and #10 report increased activity in their sensing range. These are identified as the high interest sensors and an update of the UAS sensor assignment is conducted.

Sensors #9 and #10 are incrementally weighted from 1 to 10 in step 1 of Algorithm #2. This is the high interest weighting term, w_h , mentioned in Section 3.7. Algorithm #2 is repeated and the GMM is reshaped for each successive weighting. Solutions for experimental weightings from 1 to 10 are presented. When a solution no longer varies after a third iteration or a desired WSN network configuration was reached.

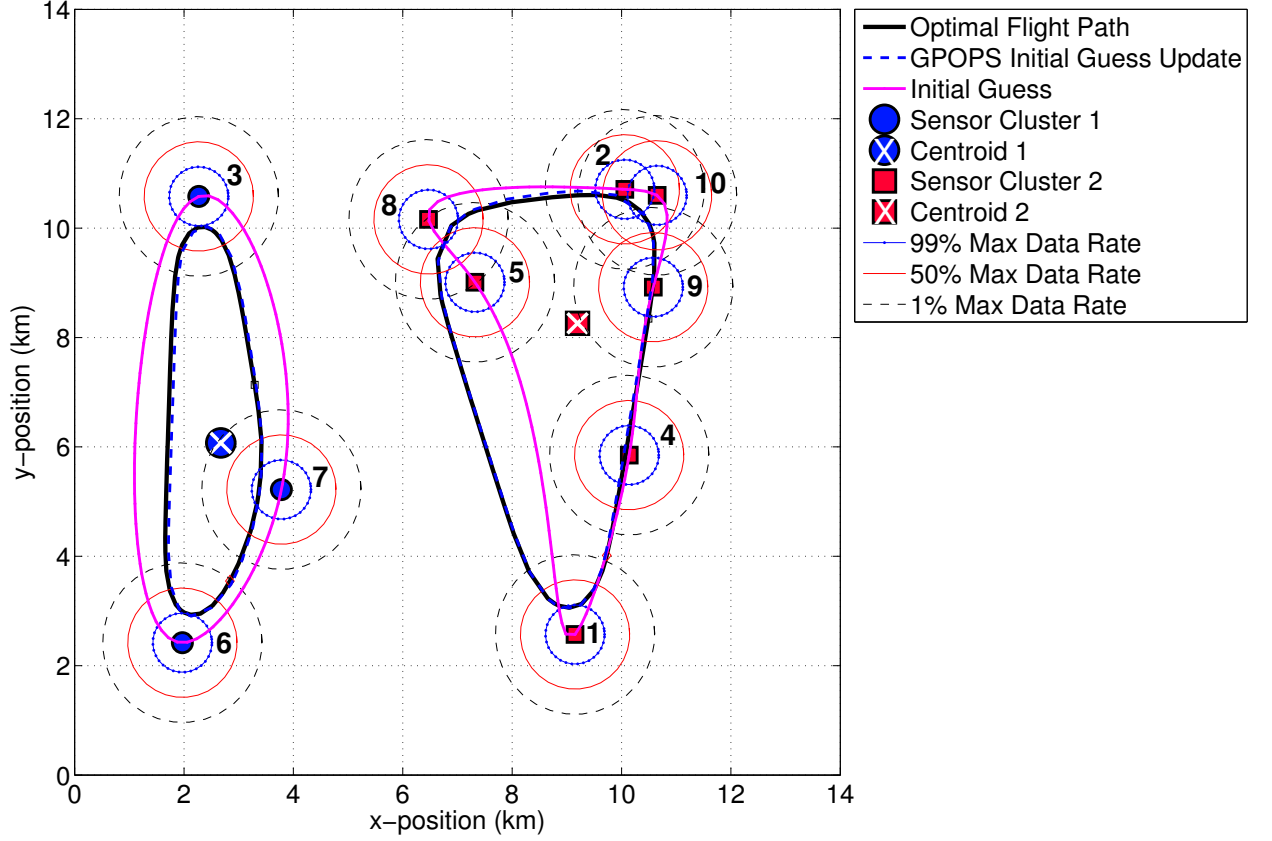
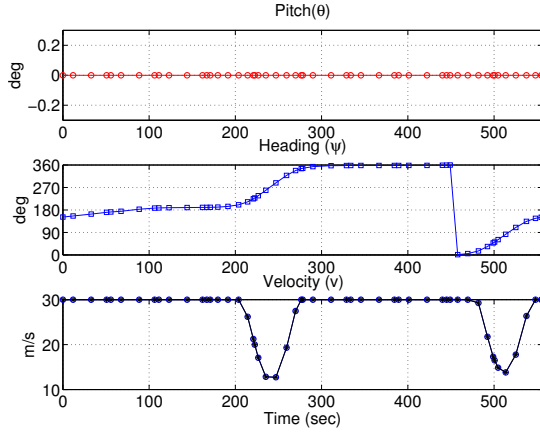


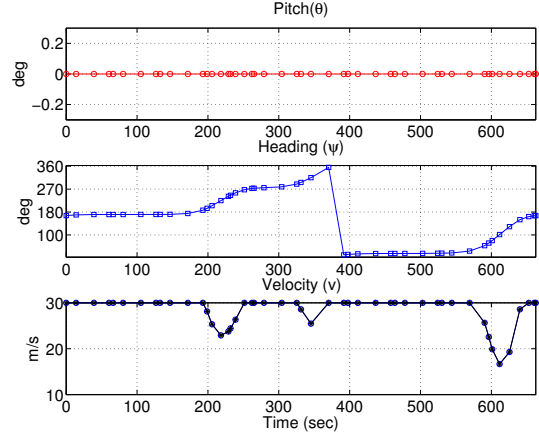
Figure 26. Ten-Sensor Reference Scenario with Initial Guess and Optimal Trajectory.

The iterations of the EM Algorithm #2 are shown in Figure 35 (note the legend is shown in Figure 29). The k -means++ centroids provided an excellent initial Gaussian means to the EM Algorithm #2. Initially the equally weighted GMM should look like the final output of the k -means++ algorithm, as seen in Figure 35d.

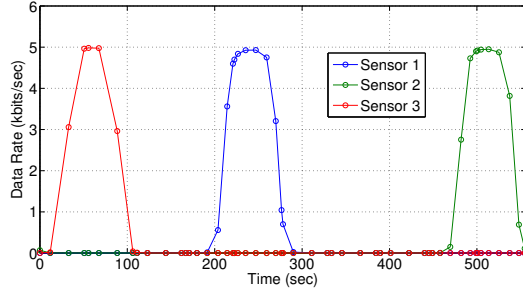
Once the desired GMM is established, the newly assigned sensors and clusters are sent back to the optimal trajectory routine for an updated optimal route. The new clusters and optimal trajectories are seen in Figure 28. As sensors start and stop experiencing relatively higher activity, this process of weighting, updating, and replanning optimal routes can be repeated as needed.



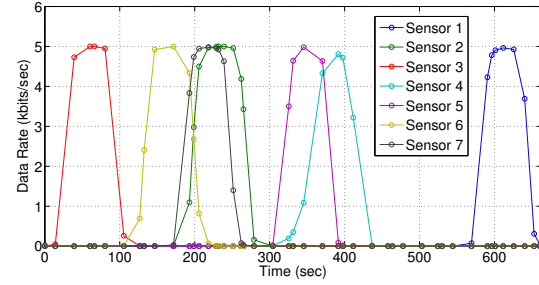
(a) UAS Controls - Sensor Cluster #1 (left)



(b) UAS Controls - Sensor Cluster #2 (right)



(c) Data Rates - Sensor Cluster #1 (left)



(d) Data Rates - Sensor Cluster #2 (right)

Figure 27. Control and Data Rates - Ten-Sensor Reference Scenario.

4.5.1 900 kbit Requirement for Sensors #6 and #10.

Under another scenario, if sensors #6 and #10 were selected, effectively choosing the two extreme points in the WSN, the EM Algorithm #2 makes no changes. This was predicted and desired. Keeping the UAS in their current assigned clusters would mean they only need to alter their airspeed or flight trajectory over the high interest sensors, and this capability was demonstrated in earlier sections. See Figure 29 of the ten-sensor re-assignment scenario for high interest sensors #6 and #10.

As in the Ten Sensor, Two UAS scenario in Section 4.4.8, UAS are required to collect at least 300 kbit of data from all the sensors. The exception however are the

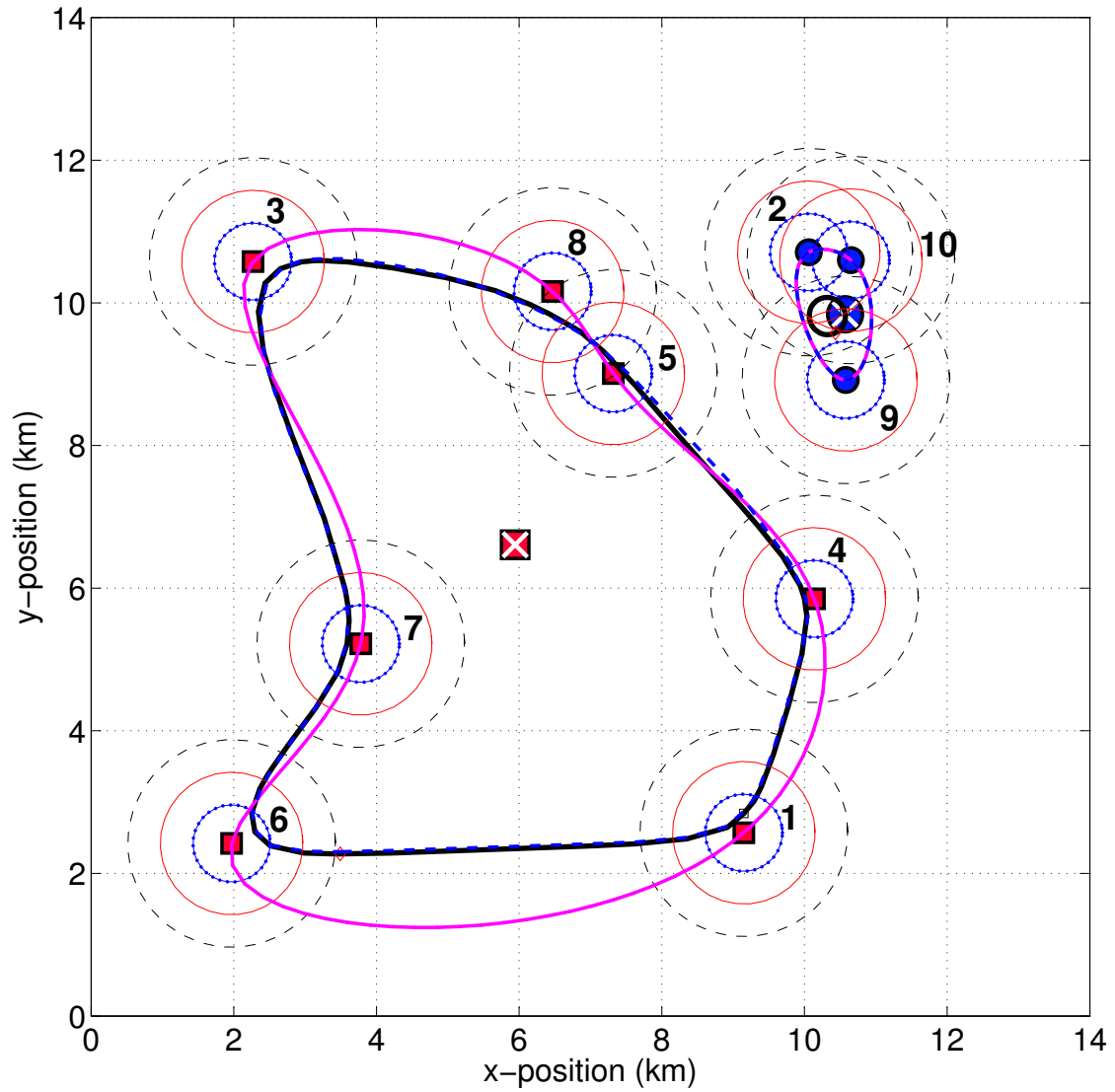
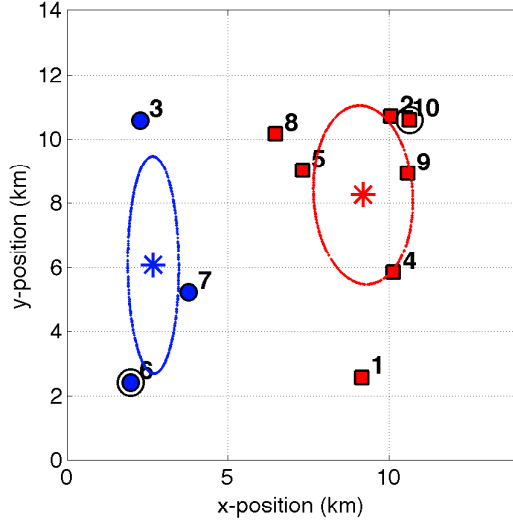


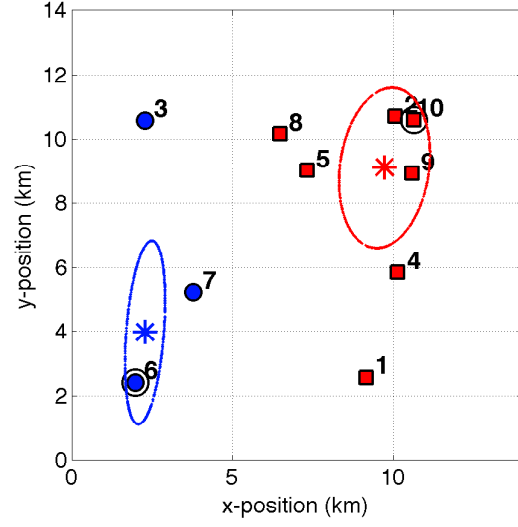
Figure 28. Ten sensor reference scenario with initial guess and optimal trajectory after sensor re-assignment.

two sensors identified as high interest earlier in the previous section. For this optimal trajectory scenario, the UAS is now required to collect triple the amount of data, or 900 kbit from sensors #6 and #10.

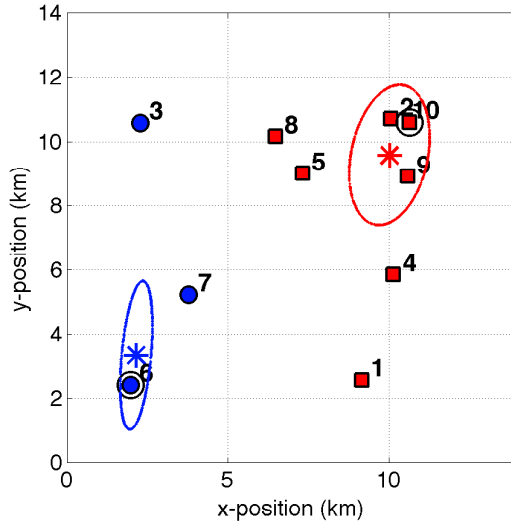
The new optimal solutions with sensors #6 and #10 as the high interest sensors and a 900 kbit data requirement is compared to the Ten Sensor, Two UAS solution from Section 4.4.8. For easier comparison, figures are shown side-by-side with pre-



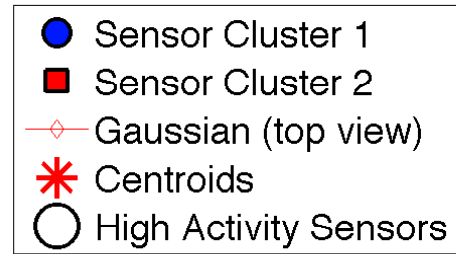
(a) $w_h=1$ on #6 and #10



(b) $w_h=5$ on #6 and #10



(c) $w_h=10$ on #6 and #10



(d) Legend for Figures 35 and 29

Figure 29. Expectation Maximization of Gaussian Mixture Model for 1, 5, and 10 weight factor on high activity sensors #6 and #10.

vious figures. The original and updated optimal trajectories are shown in Figure 30, where the trajectory can clearly be seen to track around sensors #6 and #10. But these figures only reveal part of the solution. Examining Figures 31 and 32, clear distinctions in airspeed and data rate profiles are seen between the equal data requirement trajectory in the right hand figures as compared to the left hand figures. In Figure 31b the optimal solution slowed the airspeed between the 0 and 200 second mark to collect the increased data. That decrease in airspeed is also reflected in the data rate for the same period of time in Figure 31d. The same behavior is observed in Figure 32 with the noted change in altitude as well. For the right hand side cluster optimal trajectory solution, the UAS climbed to altitude of 238 m, or 138 m above the nominal 100 m altitude. This added maneuver afforded the UAS more time within the communications range of sensor #10 to collect the required 900 kbit. If this climb maneuver is undesirable, an increase in the cost function weighing on pitch rate, β_1 , will reduce the altitude gain. This performance is similar to the single sensor scenario shown in Figures 17, 18, and 19.

Table 5 quantifies the flight time and distances flown for the Ten Sensor reference scenario with the 300 kbit and 900 kbit requirements. Examining the telemetry of the left hand cluster of Figures 30a and 30, to collect 900 kbit from sensor #6 required 1.4 km and 118 seconds more than when only 300 kbits was required.³ Similarly, Table 6 quantifies the right hand cluster of Figures 30a and 30. Collecting 900 kbit from sensor #10 required 1.4 km and 118 seconds more than when only 300 kbits was required. Although the distances increased by 9% and 4.2% for the left and right clusters respectively, the penalty for the increased data collection is overwhelming the flight time. Both left and right clusters required at least 20% more time to complete the optimal trajectory for the given increase in data requirement. It should be noted

³Percent Increase: $(|900 \text{ kbit Solution} - 300 \text{ kbit Solution}| / 300 \text{ kbit Solution}) \times 100$

that these values are subject to the cost function weights and are subject to change. If the data requirements change or UAS with differing performance are considered, the outcome is sure to also change. Ultimately, it is up to the end user on whether or not the increase in time is an acceptable trade off for the increased data collection.

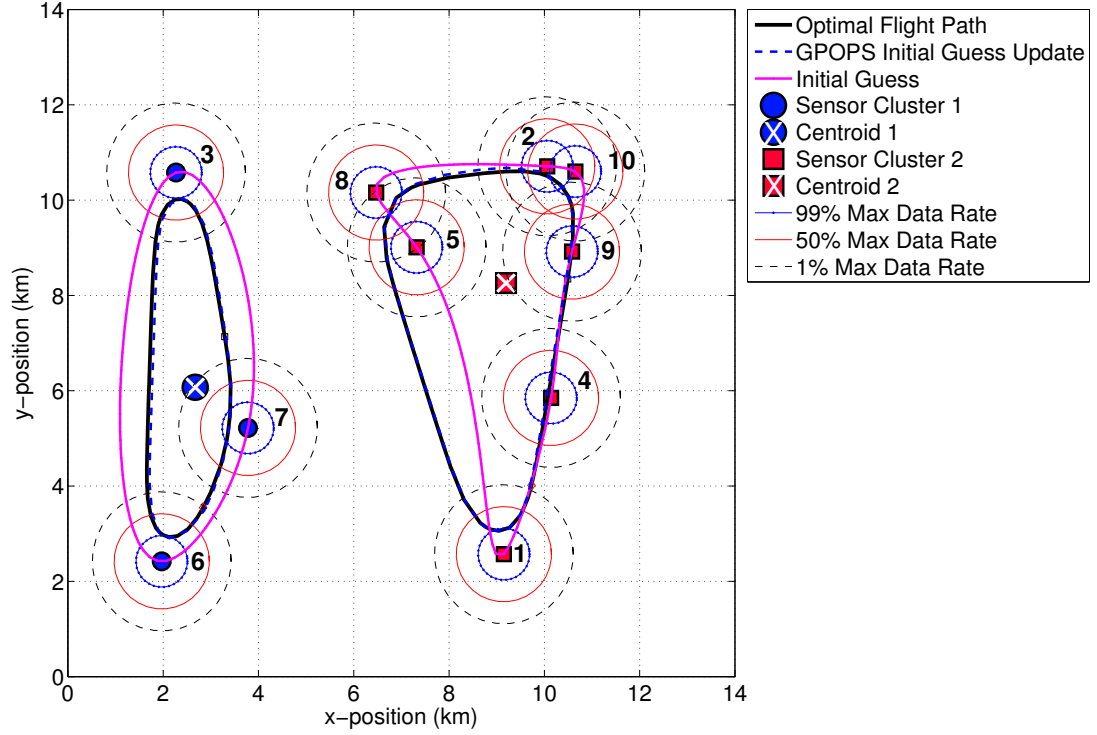
Table 5. Comparison of the flight times and distances for the Ten Sensor reference scenario for two optimal solutions. Left hand cluster only.

| Flight Time and Distance Comparison - Left Hand Cluster | | |
|--|----------------------------|--------------------------|
| Data Requirement | Distance Flown (km) | Flight Time (sec) |
| 300 kbit from All Sensors | 15.5 | 554.8 |
| 900 kbit from Sensor #6 | 16.9 | 673.6 |
| Increase (%) | 1.4 (9%) | 118.8 (21.4%) |

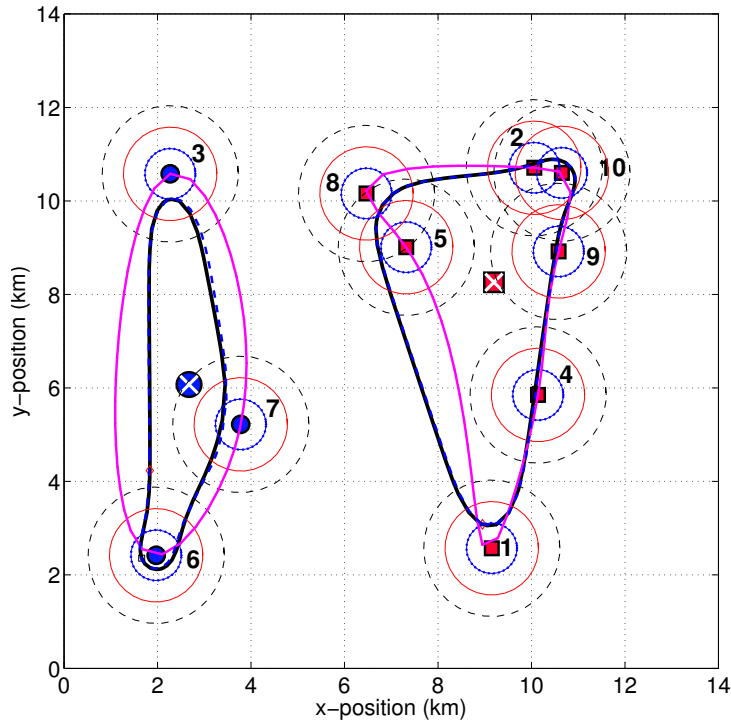
Table 6. Comparison of the flight times and distances for the Ten Sensor reference scenario for two optimal solutions. Right hand cluster only.

| Flight Time and Distance Comparison - Right Hand Cluster | | |
|---|----------------------------|--------------------------|
| Data Requirement | Distance Flown (km) | Flight Time (sec) |
| 300 kbit from All Sensors | 19.1 | 652.4 |
| 900 kbit from Sensor #10 | 19.9 | 787.2 |
| Increase (%) | 0.8 (4.2%) | 134.8 (20.7%) |

Finally, the data collected from the two optimal solutions is seen in Figure 33. The black bars illustrate the data collected by the UAS when the data collection requirement was set at 300 kbit per sensor. The orange bars illustrate the data collected by the UAS when the collection requirement for sensors #6 and #10 was increased, while maintaining the minimum 300 kbit from the remaining sensors. This figure illustrates and confirms the proper amount of data was collected from each sensor in both scenarios.

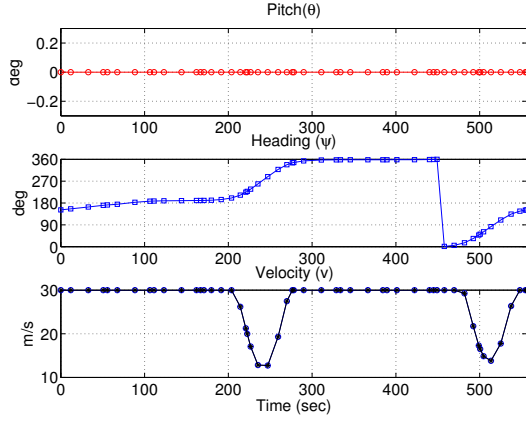


(a) Ten sensor reference scenario solution for 300 kbit of data from all sensors.

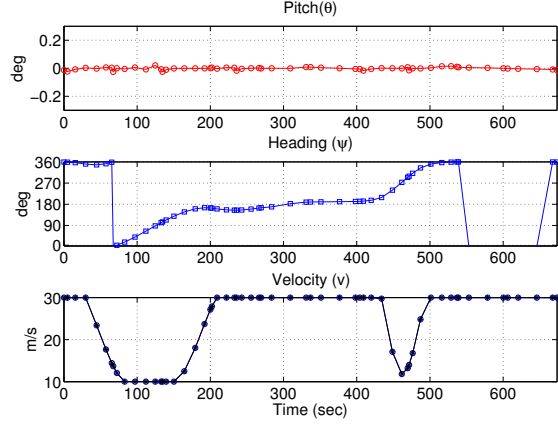


(b) Ten sensor reference scenario for 900 kbit of data from sensors #6 and #10.

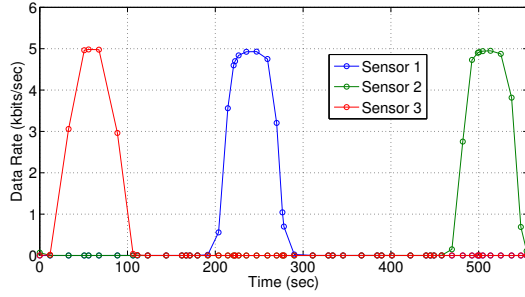
Figure 30. Ten-sensor reference scenario comparison between equal data requirements and increased data requirements from sensors #6 and #10.



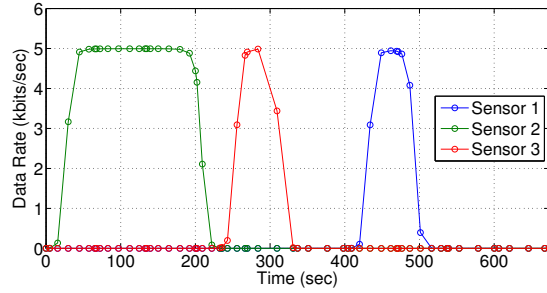
(a) UAS controls for sensor cluster #1 (left). 300 kbit required from all sensors.



(b) UAS controls for sensor cluster #1 (left). 900 kbit required from sensors #6 and #10.

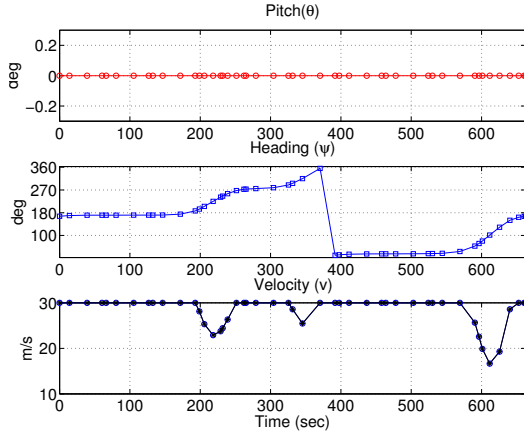


(c) Data rates from sensor cluster #1 (left). 300 kbit required from all sensors.

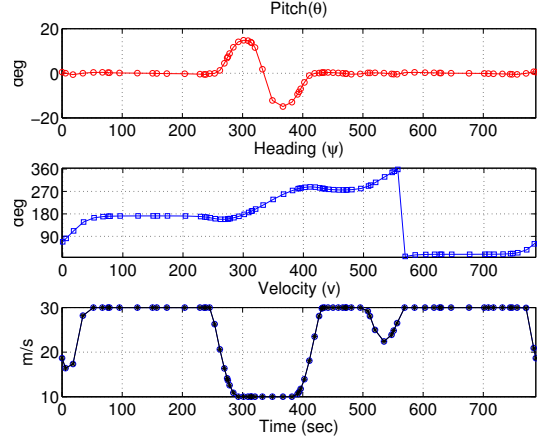


(d) Data rates from sensor cluster #1 (left). 900 kbit from required sensors #6 and #10.

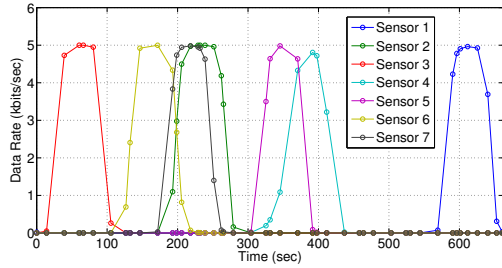
Figure 31. Control and data rates comparison for the ten sensor reference scenario, left hand cluster of Figure 30 only. (a) and (c) are reproduced from Figures 19 and 18, and represent optimal solutions for equal data requirements for 300 kbit from all sensors. (b) and (d) illustrate the change in control and data rate to collect 900 kbit of data from sensor #6 only.



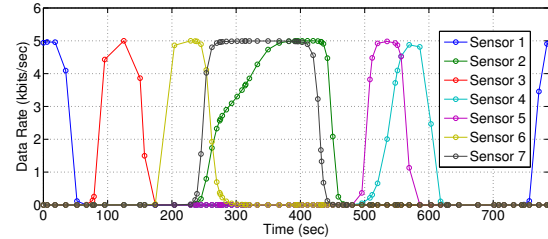
(a) UAS controls for sensor cluster #1 (right).
300 kbit required from all sensors.



(b) UAS controls for sensor cluster #1 (right).
900 kbit required from sensors #6 and #10.



(c) Data rates from sensor cluster #1 (right).
300 kbit required from all sensors.



(d) Data rates from sensor cluster #1 (right).
900 kbit from required sensors #6 and #10.

Figure 32. Control and data rates comparison for the ten sensor reference scenario, right hand cluster of Figure 30 only. (a) and (c) are reproduced from Figures 19 and 18, and represent optimal solutions for equal data requirements for 300 kbit from all sensors. (b) and (d) illustrate the change in control and data rate to collect 900 kbit of data from sensor #10 only.

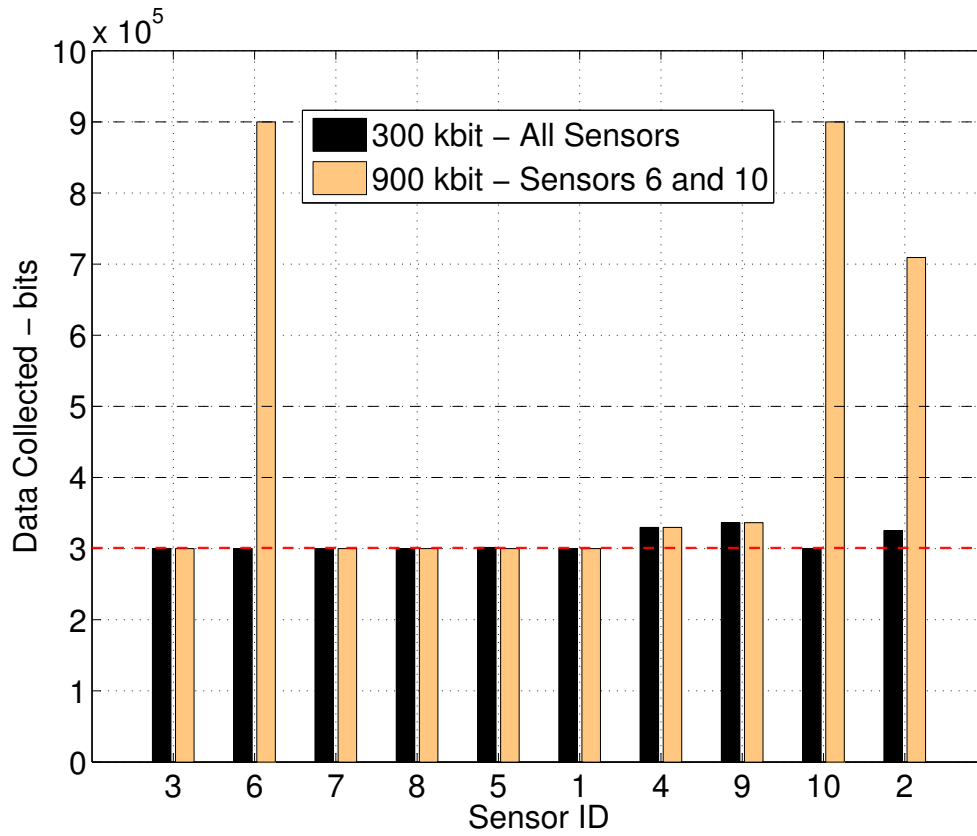


Figure 33. Ten sensor reference scenario comparison of data collected from two optimal solutions. The black bars indicate data collected with a 300 kbit minimum for all sensors. The orange bars show an increased data collection from sensors #6 and #10 respectively, while maintaining 300 kbits from the remaining sensors.

As a result of the selecting sensors #6 and #10, there was no change in the clustering of the sensors and thus no change in UAS assignments. In this case, the optimal solution required altering the trajectory, lowering the airspeed, and increasing altitude temporarily to collect the new data requirements. The optimization routine took advantage of the vertical solution space to remain within the sensors communications range longer. Although this optimal solution increases time and distance, it also ensures meeting the minimum required data.

4.6 Simulation

High fidelity simulations are the next logical step to verify the optimal trajectory solutions. The **A**ir **V**ehicle **T**est **A**nalysis and **S**imulation **M**ulti-**A**gent **S**imulation **E**nvironment (AMASE) suite simulates the UAS flying the calculated optimal trajectory [127]. This section describes AMASE and presents the simulation results.

Understanding how AMASE works requires knowledge of the foundational software architecture. The **C**ommon **M**ission **A**utomation **S**ervices **I**nterface (CMASI) defines the underlying protocol that allows all systems to interface seamlessly [127].

AMASE is a simulation toolset for the analysis and demonstration of UAS mission technologies. AMASE includes the necessary components to create scenarios, simulate UASs and some on-board components, and interact with control algorithms to command UAS in a scenario. AMASE makes extensive use of CMASI⁴ based on work in [127].

The CMASI data structure enables data objects to be shared and accessed over a network and is agnostic to the system infrastructure. Within the protocol, information is packaged into a message using an Extensible Markup Language (XML) format. The information contained in these messages ranges from high level items like mission tasks and operational constraints, down to waypoint commands and payload control. As long as messages have the requisite CMASI data structures, the information contained in the messages can be interpreted and executed. Using CMASI allows for multiple systems to effectively interact and share data over a common network. Although the type of messages a particular system subscribes to may differ depending on their role, the interface remains consistent over the various platforms.

AMASE and CMASI are an all-in-one simulation environment to perform both software and hardware in the loop testing. AFRL's Controls Science Branch uses

⁴Extracted from AMASE User's Guide provided by AFRL

the AMASE platform to verify UAS control algorithms, mission objectives, and system interoperability, prior to actual flight test⁵. A strong correlation exists between simulation and real-world flight test success due to the robustness of AMASE.

For this research, AMASE is an intermediate step prior to flight test. Employing AMASE in this manner has several benefits. The first ensures numerical results are properly translated into CMASI command messages with pertinent information for the UAS to fly the desired trajectory. Without properly translating the numerical solution into the correct XML formatted messages, the optimal trajectory cannot be flown in AMASE or transmitted to an airborne UAS. When the trajectory is flown properly in AMASE then the UAS command messages are correctly formatted and ready for flight test. Second, data collected from simulation is analyzed and evaluated independent of MATLAB[®]. This separate evaluation verifies the MATLAB[®] results, verifying the optimal trajectories. Lastly, AMASE allows for a variety of simulations in a representative environment.

4.6.1 Simulation Procedure.

The following scenario takes place at Camp Atterbury Training Center, Indiana, and a future flight testing site. Due to access and airspace limitations, the sensors are spatially closer together than desired, but nevertheless provide valuable insight into the data collection scheme. To begin, a TSP solution is flown in AMASE and a minimum of 730 kbits were collected from each sensor. The TSP solution represent traditional flight planning and flight trajectories. The same sensor locations were entered into GPOPS-II and the minimum amount of data required was set to 730 kbits for each sensor. An optimal flight trajectory was calculated and entered back into AMASE for comparison.

⁵Based on discussions with AFRL Controls Science Branch engineers

The collocation points produced by GPOPS-II are used to input the optimal flight trajectory into AMASE. The collocation points along the optimal trajectory contain the UAS position, altitude, and airspeed information, making them ideal waypoints for the UAS. A corresponding time vector is also included for reference. Also, the UAS x and y positions from the numerical solution are converted to latitude and longitudinal coordinates to align with the GPS navigation system of the on-board autopilot. This is done using one of the sensors with known latitude and longitude coordinates as an origin. The conversion is based on the 1984 World Geodetic System standard. After AMASE simulations are completed, XML formatted messages include a unique numbered waypoint for each collocation point from the GPOPS-II solution. For each unique waypoint, XML command messages included UAS airspeed in meters per second, altitude in meters, latitude and longitude in decimal degrees, and the next waypoint. Furthermore, the simulated UAS in AMASE is commanded by the exact same XML messages transmitted to the actual UAS in flight test. This seamless transition from simulation to flight test is one of the main reasons this process is conducted.

During flight test, the UAS flies a fixed altitude and airspeed. To match flight test conditions, a fixed altitude of 152.4 m and velocity of 15 m/s were programmed into the optimal trajectory solution and AMASE. This is a variation from the theoretical approach in Chapters III and IV, however necessary to match the test conditions. As in the earlier test cases, the maximum data rate, distance, and stiffness were set to 5 kbits per second, 1 km, and 0.01 respectively. Figure 34a compares the optimal and TSP trajectories. Figure 34b compares the total data collected at each sensor via the numerical, AMASE, and TSP trajectories. All three models use the empirical communication model approximated by the sigmoid function.

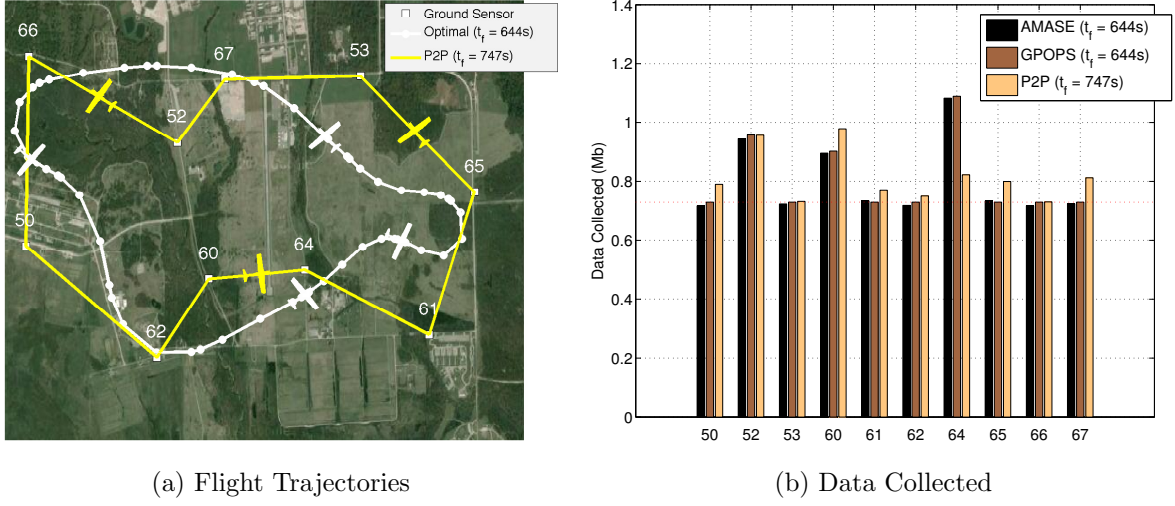


Figure 34. Comparison of a TSP and optimal trajectory collecting 730 kbit of data.

4.6.2 Simulation Results and Conclusions.

Figure 34a illustrates the TSP (also referred to as “P2P” for point-to-point) and optimal trajectories flown in AMASE. The optimal trajectory is nearly 100 seconds faster, constituting a 14% savings in flight time while meeting the minimum data collection requirements.

Two conclusions are made from Figure 34b. First, data collected from the optimal trajectory in both GPOPS-II and AMASE are nearly identical. This result makes sense because the waypoints and the communication model used for both the MATLAB[®] and AMASE environments are the same. Achieving near identical data collection is important as it verifies the optimal trajectory solved using GPOPS-II is accurately translated into the XML format and simulated in AMASE. The small differences between the two are attributed to the different integration methods (Gaussian quadrature vs Runge-Kutta methods) used for calculating the total data collected. Second, the overall data collected from the optimal trajectory achieves the minimum requirement for each ground sensor and has comparable data collection to the more

traditional TSP route. The optimal trajectory satisfied the scenario requirements without degrading performance.

Waypoint following is the most common flight trajectory planning method for unmanned aircraft. The collocation points in the optimal solution produced by GPOPS-II contain enough state information to act as the waypoints. Simply put, the UAS will fly each of the collocation points since they contain position, altitude, and airspeed information. With adequate simulation, improvements and refinements can be made to the mathematical models prior to any future flight test.

4.7 Chapter Summary

In this chapter, the specific problem setup details and software settings were presented, followed by various scenarios exercising the prescribed methodologies from Chapter III. Results present support the methodologies as feasible methods capable of solving the complex challenge of collecting data in WSN. Based on a computer simulation of the Atterbury scenario using the optimal trajectory, a 14% savings in flight time was realized over the traditional TSP path. The next chapter explores one additional scenario with the added realism of flight test for comparison.

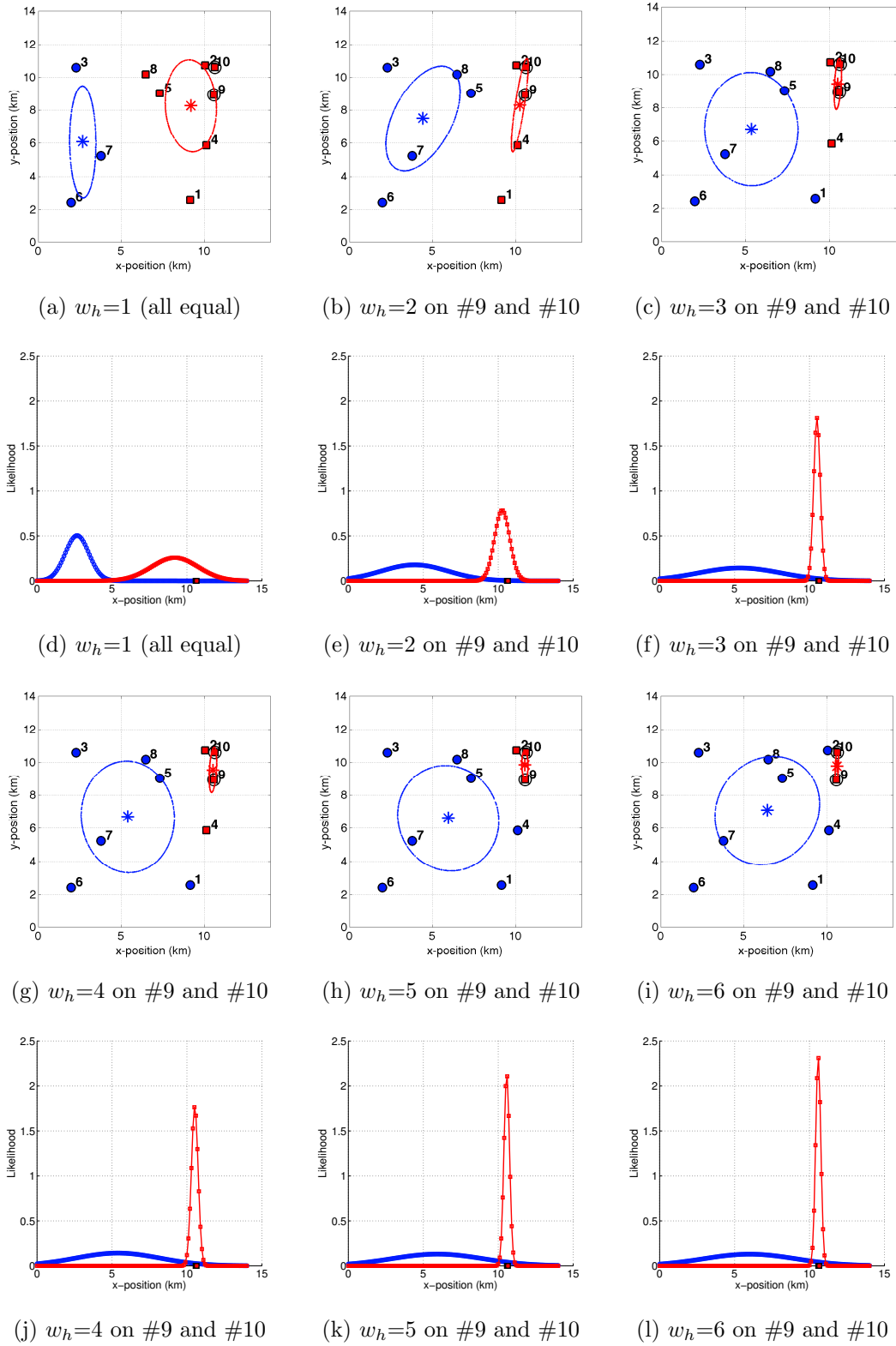


Figure 35. EM of GMM for 1-6 weight factor on high activity sensors #9 and #10. Figures (a),(b),(c),(g),(h), and (i) represent a top down view of the Gaussians with incremental weighting on the high activity sensors #9 and #10. Figures (d),(e),(f),(j),(k), and (l) represent the x-axis view (side view) of the Gaussians for increasing weight factors.

V. Flight Test

5.1 Introduction

In 2014 and 2015, proof of concept flight tests were conducted to validate that optimal trajectories would collect a minimum amount of data from static ground sensors. AFRL and their contractor Wright State Research Institute (WSRI) are currently conducting research with UAS operating in WSN. AFRL, who sponsored this dissertation, in coordination with WSRI conducted all aspects of the flight testing, including the organization, funding, necessary permissions, training, equipment, support, and operations. This section outlines the purpose, equipment, procedures, testing, results, analysis. Conclusions from the flight test are drawn and recommendations for future tests are presented.

5.2 Purpose

First, flight test helped to characterize hardware performance, specifically the range and reliability of the communications link between the UAS and the ground sensors. This step enabled development of the empirical communications model used in Chapters III and IV. Second, flight testing helps validate the numerical solution as sufficient enough to collect a specified amount of data from ground sensors. After a nominal communications range was determined from the first set of tests, optimal trajectories were calculated and flown in a second round of tests.

5.3 Equipment and Range

The following sections briefly describe the aircraft, autopilot, control and base stations, and ground sensor used during the flight tests.

5.3.1 Aircraft.

AFRL and WRSI employed the Procerus Unicorn flying wing. The catapult launched Procerus Unicorn with Piccolo autopilot, affectionately named the “*P-corn*”, is an expanded polypropylene foam aircraft with a 1.83 meter wingspan, weighing approximately 4.54 gross kg, but can vary depending on payload. Utilizing a brushless direct current motor and a nominal configuration of six 6-cell fully charged lithium polymer batteries, it has a flight time of approximately 1.5 hours. The aircraft has a stall velocity of 7.2 m/s, and flies between 14 and 27 m/s.

5.3.2 Autopilot and Control Stations.

The P-corn employs the 110 gram Cloud Cap Piccolo SL autopilot system. For command and control, the Piccolo SL utilizes a 900 Mhz frequency with a 1Hz update rate. For data communications, WiFi data links use a 2.4 GHz signal. The control station is part of a base station of hardware and software utilized by AFRL and WSRI. The base station for flight tests includes a mobile control truck, tracking antenna, catapult launch vehicle, spare parts, and more. The control station suite incorporates Cloud Cap’s native control station married with specialized equipment dedicated to UAS flight testing. Figure 36 shows a screen shot of the control station and photographs of aircraft preparation, tracking antenna, and general base station layout.

5.3.3 Ground Sensors.

Figure 37 shows a typical unattended ground sensor used by WRSI and AFRL. All ground sensors are manually assembled and share identical configurations. They employ doppler radar for detection, global positioning system and WiFi antennas. The sensors are operated by Gumstix microcomputer with a Linux based operating

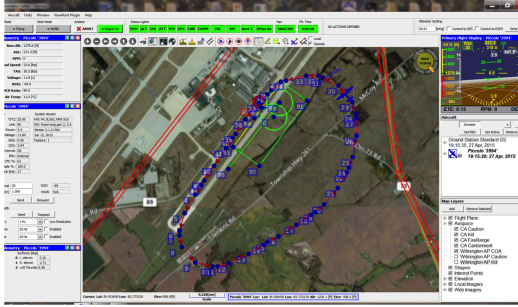


Figure 36. AFRL and WSRI control station screenshot, flight test hardware, and operations setup.

system. Aircraft and ground sensors are directly addressed via an internet protocol 802.11b communication architecture. Communication messages are passed using the ZeroMQ messaging library.

5.3.4 Test Ranges.

Testing was conducted at two locations. Ground testing was first conducted at Wright-Patterson AFB, Area B runways. The ground testing was largely used to ensure connectivity of the ground sensors to a mobile laptop acting as a UAS. Wilmington Air Park (ILN) in Wilmington, Ohio hosted the flight testing. Wilmington Air Park is within a short 45 minute drive from AFIT and AFRL and proved to be an excellent site for small scale testing. Although limited in airspace, initial proof of

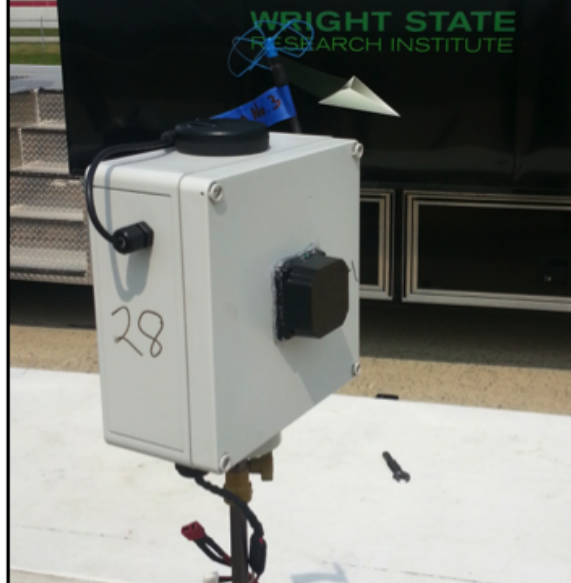


Figure 37. AFRL WRSI Ground Sensor.

concept flights were conducted here. The flight test team utilized the 2,743 m 4R/22L runway and the surrounding land to place the ground sensors, UAS launch and recovery, and base station operations. The flight test was conducted with permission from the FAA under a Certificate of Authorization obtained by AFRL. Wilmington Air Park is 382.2 m above sea level.



Figure 38. Wilmington Air Park, Wilmington, Ohio.

5.4 Experimental Procedure

Ahead of the actual experiment are countless years of collective experience feeding into each flight test. Prior to the procedure below, both AFRL and WRSI have completed hardware and software preparation, bench level testing, aircraft modifications, sensor preparation, administrative and legal permissions, airfield and airspace coordination, and countless other requirements. The following list relates only to this research and cannot be completed without all the aforementioned actions. Steps #3 and #4 are detailed in Section 4.6.1 and only summarized here.

1. Determine airspace boundaries, place and activate sensors.
2. Calculate optimal trajectory - Input sensor locations, minimum data required, aircraft parameters, and mean wind speed and direction.
3. Simulate in AMASE - Transfer the optimal trajectory to AMASE for HITL and SITL testing (optional).
4. Translate optimal trajectory into XML command message format and upload to airborne aircraft.
5. Aircraft executes commanded flight trajectory and collects data from sensors.
6. Download and analyze data and telemetry after completed orbits of the WSN (completed while UAS still in flight).

5.5 Settings and Values

The settings and values in Table 7 were determined through trial and error in simulation and prior flight tests. Airspeed and altitude settings are subject to aircraft type and airspace boundaries. The autopilot uses inner-loop throttle commands to

Table 7. Selected settings for flight test.

| | |
|---|---------------|
| Commanded True Air Speed | 20 m/s |
| Commanded Altitude | 100 m |
| Fixed Bit Rate from each Sensor | 7.85 kbit/sec |
| Minimum Data Required from each Sensor | 750 kb |
| Maximum Assumed Comm Range (Days 1 and 2) | 1 km |
| Maximum Assumed Comm Range (Day 3) | 1.15 km |

track to a given altitude, meaning the velocity varies as the UAS attempts to fly at a set altitude. Subsequent airspeed commands are subordinate to aircraft altitude commands. For the data collection test, the ground sensors were programmed to continuously transmit a 785 bit sized message at a 10 Hz rate. Once the aircraft was within range and connected to the ground sensor, it received and recored data messages.

After reviewing telemetry from flight test days one and two, the communications range for day three was increased slightly in an attempt to match the numerical and simulated data collected.

5.6 Results

Over the course of three days, over 40 orbits of the WSN were flight tested at Wilmington. Days one and two were flown with an assumed maximum communication range of 1 km and day three was flown with an assumed maximum communication range of 1.15 km. This section presents the data collected for all three days and specifically focuses on the flight times and distances flown on day three.

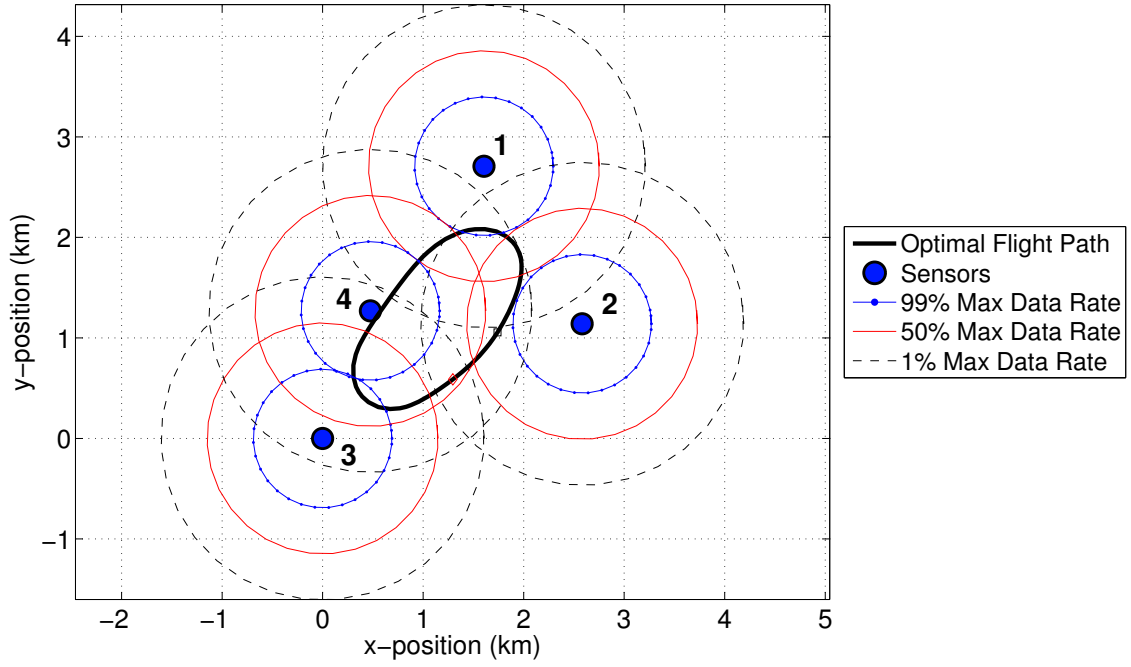


Figure 39. Numerical solution (GPOPS-II) for Wilmington sensor layout with 1.15 km assumed comm range.

5.6.1 Data Collected.

Figures 40 and 42 illustrate a comparison between the total data collected from the numerical (GPOPS) solution, simulation (AMASE) solution, and the flight test data. Since data collected from flight test varied from orbit to orbit, the mean and standard deviation from multiple orbits are used to better illustrate the results. The minimum required data is a dashed line indicated by the note, “*Min Data Req’d: 750 kb*”. Data was collected from four sensors placed around the Wilmington Air Park.

Figure 40 shows the total data collected from flight test days one and two. Since the communications range was set to 1 km for both days, their results are presented together. Recall the simulation of the optimal path in AMASE uses a discrete communications model. In other words, it connects and disconnects exactly at the specified maximum communications range. In contrast, the empirical communications

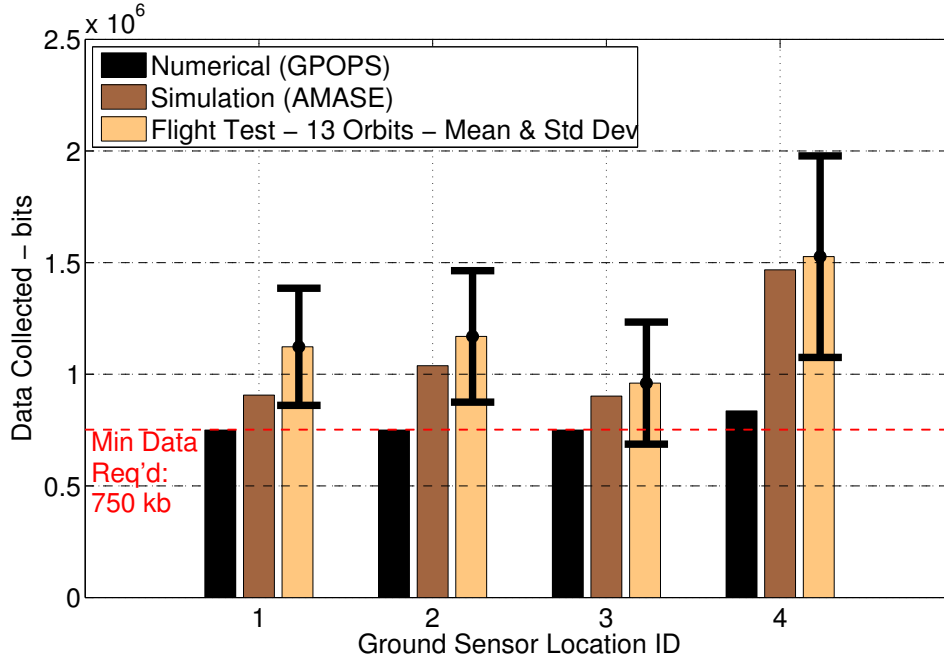


Figure 40. Flight test data collected on days one and two at Wilmington.

model starts to slope off at approximately 0.5 km following the sigmoid function (and corresponding stiffness value). The excess data collected in the flight test indicated the aircraft remained well within communications range of the sensor. Although on average, the 1 km communications model collected more than the minimum required, the amount collected on each orbit varied as indicted by the standard deviation bars. Ideally, the lower end of the standard deviation is above the minimum required data line to ensure that all the orbits can meet the collection criteria. This secondary minimum condition is true in Figure 40 for all the sensors except sensor #3, but is still very close.

Figure 42 also represent data collected, but from flight test day three using a longer communications range of 1.15 km seen in Figure 41. In contrast to the orbit of the 1 km communications range, the increased comm range results in a spatially smaller orbit, requiring less distance to and time to complete. As expected, the mean

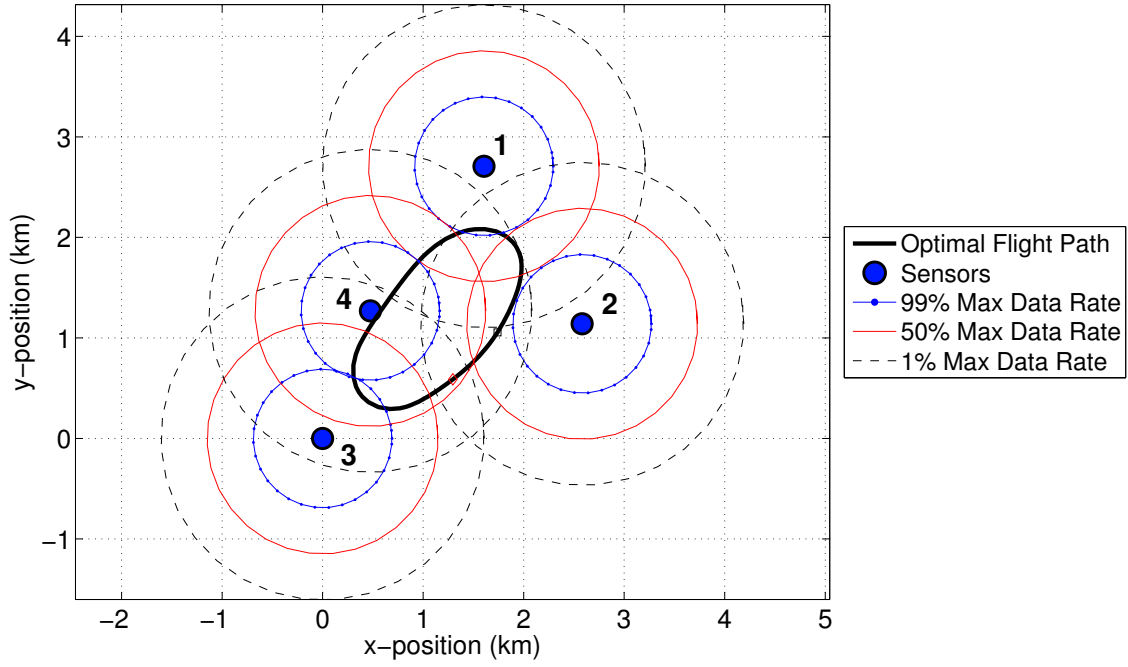


Figure 41. Numerical solution (GPOPS-II) for Wilmington sensor layout with 1.15 km assumed comm range.

of the data collected with a 1.15 km range is lower than for the 1 km communications model flights. However, the variation of the data collected on each orbit is more variable than in the previous flights. Although the mean data collected is very close the minimum required for sensors #1 and #3, their lower end standard deviation drops far below the minimum data required. Sensor #2 had a significant drop in data reported, more than likely due to longer distance incurred as a result of the 1.15 km communications range.

5.7 Day Three Flight Time and Distances Flown Comparison

A further look at the telemetry from flight day three helps to provide some insight on performance of the aircraft. Table 8 lists the variations in true, ground, and wind speeds provided better insight of the flight environment for that specific day. A change

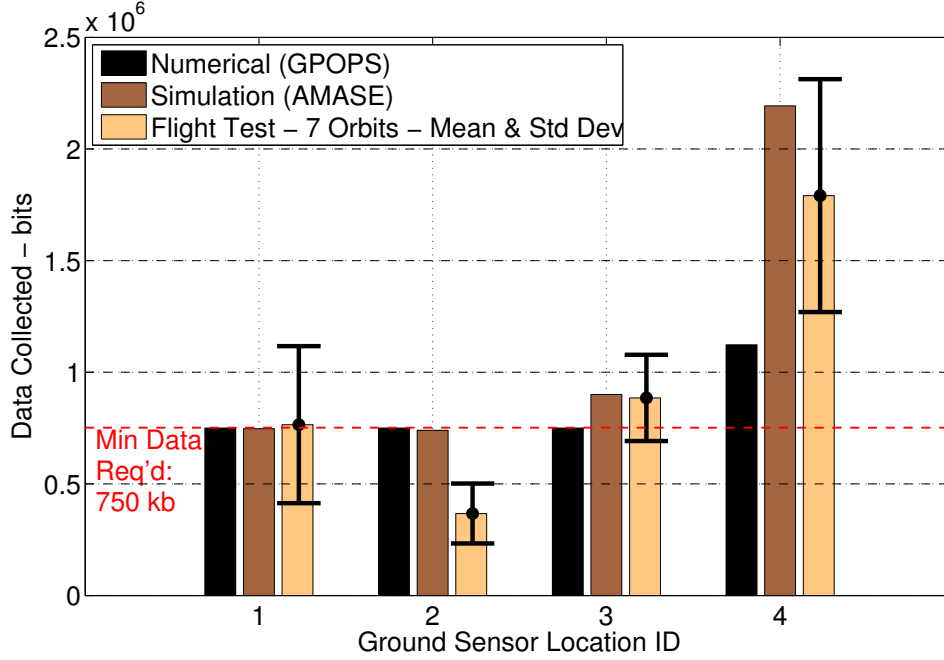


Figure 42. Flight test data collected on day three at Wilmington.

in ground speed along the optimal trajectory effectively changes the data rates as the UAS is either in a sensors communications range for a longer or shorter period. For this particular scenario, visiting each sensor in a traditional TSP fashion is expected to take approximately 388 seconds. Collecting the minimum of 750 kb requires 263 seconds, or a difference of 125 seconds, approximately 32%¹. In flight test, the same optimal flight path required 276 seconds. Despite the variations in aircraft speed and wind effects, the difference between the numerically calculated optimal flight time and the actually flight time is only 13 seconds or 4.8%². The values are listed in Table 9. Similarly, the distance required by the TSP versus the optimal showed significant savings. Table 10 lists the distances flown. The optimal trajectory resulted in a 2,585 m shorter flight path when compared to the TSP route. And in flight test, the aircraft flew nearly the exact route, varying only by 0.1%. The flight time and distance were

¹Percent Savings: $|TSP - Optimal|/TSP \times 100$

²Percent Difference: $|TSP - Optimal|/(0.5 \times (TSP + Optimal)) \times 100$

Table 8. Selected telemetry from flight test day three. As reported by the on-board Piccolo autopilot from seven successive orbits of the WSN.

| Day 3 Telemetry | Minimum (m/s) | Mean (m/s) | Maximum (m/s) |
|--------------------------|------------------|---------------|------------------|
| True Air Speed | 16.9 | 20.2 | 26.0 |
| Ground Speed | 10.2 | 19.0 | 30.0 |
| Winds Aloft (South East) | 2.2 | 5.9 | 9.6 |

Table 9. Comparison of flight times between numerical solution and flight test results for a maximum assumed communication range of 1.15 km.

| Flight Trajectory Type | Time |
|--|-----------------|
| TSP (Calculated) | 388 sec |
| Optimal (Numerical Solution) | 263 sec |
| Time saved Optimal vs TSP (Percent Savings) | 125 sec (32.2%) |
| Mean Flight Test Time of Optimal Trajectory (7 Orbits) | 276 sec |
| Numerical Solution vs Flight Test - Percent Difference | 4.8% |

flown accurately, meaning, they were flown as calculated and commanded. Wind effects are also likely to be a factor as the ground speed varied from 10.2 m/s to 30 m/s.

5.8 Day Three Communications Analysis

Communication anomalies observed and this section examines some communications parameters from sensor three on flight day three. Figure 43 illustrates the trajectories calculated and flown on day three. Only orbit #3 is shown shown for brevity and clarity, but all orbits shared similar results. The solid line indicates the numerically calculated optimal path given a 1.15 km assumed communications range. The smaller red circles indicate every moment the aircraft receives a data message

Table 10. Comparison of distance flown between numerical solution and flight test results for a maximum assumed communication range of 1.15 km.

| Flight Trajectory Type | Distance Flown |
|--|-----------------|
| TSP (Calculated) | 7,853 m |
| Optimal (Numerical Solution) | 5,268 m |
| Time saved Optimal vs TSP (Percent Savings) | 2,585 m (32.9%) |
| Mean Flight Test Distance Flown of Optimal Trajectory (7 Orbits) | 5,274 m |
| Numerical Solution vs Flight Test - Percent Difference | 0.1% |

from any of the sensors, and is plotted at the latitude, longitude, and altitude when a particular message was received. An unexpected gap in the communication exists as the aircraft passes sensors #1 and #2. This communication gap is attributed to hardware anomalies and discussed further in Section 5.9.

5.8.1 Day Three, Sensor Three Communciations Analysis.

Further investigation into the communication telemetry reveals an inconsistent receive rate by the UAS. Zooming in and examining the lower portion of the optimal trajectory from Figure 43, the small red circles indicate message data received by the UAS at that position from sensor location #3 and is shown in Figure 44. At first glance, the UAS was late to connect to the sensor and a gap is evident. Although well within range, the UAS did not connect until approximately 0.6 km away from the sensor, as indicated by the first received data message. Despite the sensor consistently transmitting messages at a 10 Hz rate, the UAS often received buffered messages, and at various receive rates. Figure 45 shows a relationship between the distance from the sensor at location #3 and the UAS. It also shows the time between messages averages 0.1 seconds (10 Hz), however it is not consistent throughout. Clearly, at approximately the 39 and 44 second mark, a gap is visible and a delay in message

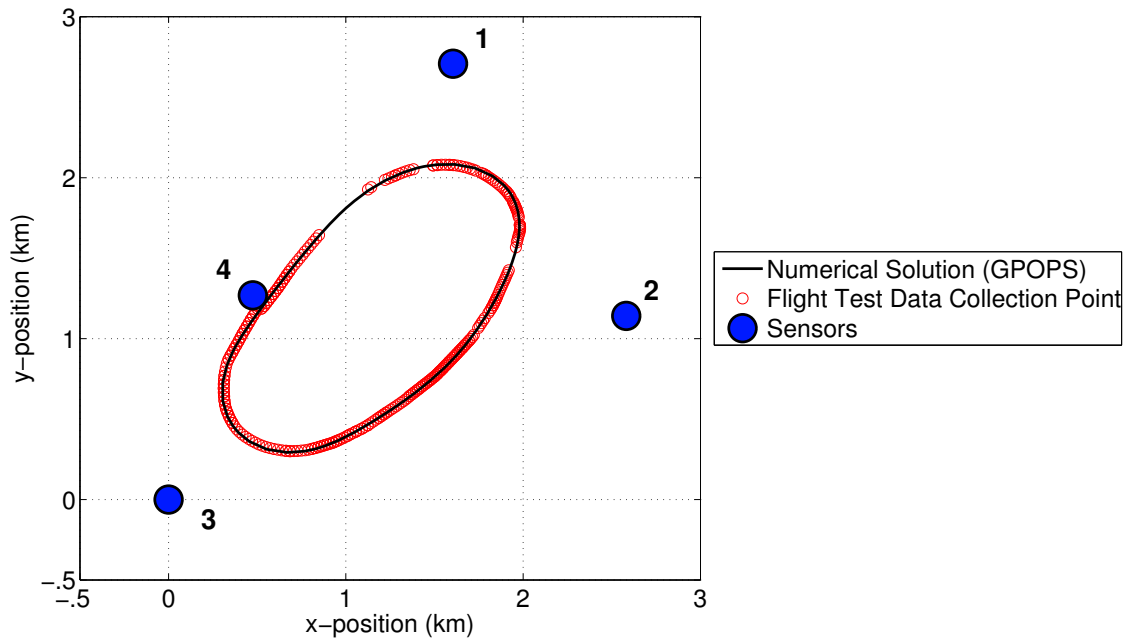


Figure 43. Numerical solution (GPOPS-II) and flight test telemetry of day three orbit three for a 1.15 km assumed comm range.

receipt is recorded. In most cases however, messages were not dropped but delayed by 2 or more seconds, often held in the sensor's buffer. A further examination of the telemetry showed no messages were lost. So despite being buffered, all the messages that were sent, were eventually received by the UAS. On another orbit, a 40 second delay was recorded before buffered messages were received. In a few other orbits, messages were buffered long enough to be dropped from the buffer and not transmitted to the UAS.

5.9 Post Flight Discussion

Overall, the optimal trajectory successfully collected a minimum amount of data, albeit not perfectly, and in some cases more than the required amount. Although there exist some degrading factors, the aircraft flew the calculated optimal trajectory accurately and the UAS communicated with the ground sensors, collected data, and

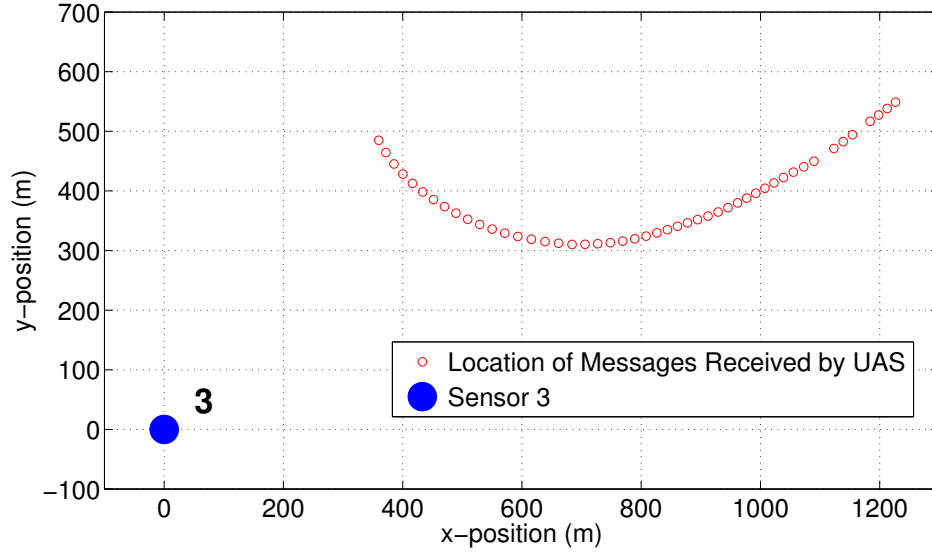


Figure 44. Sensor #3’s unique messages received at the UAS along the optimal flight trajectory. Gaps indicate buffered or dropped messages. For orbit #3, sensor locaiton #3.

proved the viability of the method. As expected in any flight test experiment, a host of typical and atypical challenges arose. The following paragraphs address possible degrading factors, challenges, and makes recommendations for future testing.

5.9.1 Communications Anomolies.

The flight test team speculated on the volatility of the communications between the UAS and sensor. Since the UAS and ground sensors are low-powered devices compared to the airport environment, it is likely they may be more susceptible to radio frequency interference than a typical commercial aircraft.

First, the Wilmington Airpark is an active airport with higher than average radio frequency activity. Some likely sources of interference included the power lines passing in and out of the test range, airport radars and radios, and shielding effects from trees (the direct line of sight between the UAS and some ground sensors passed through

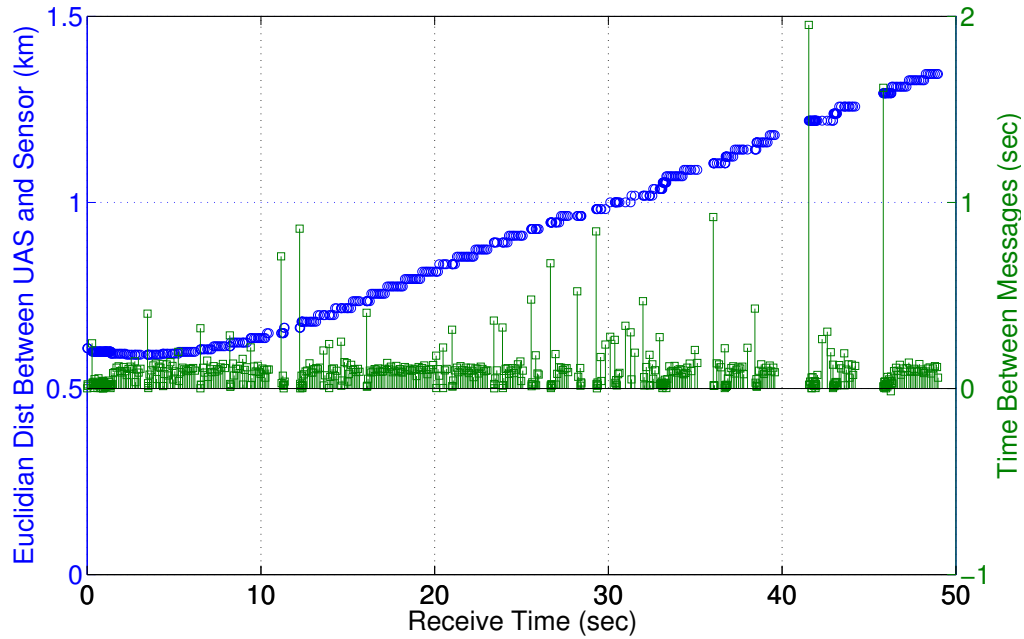


Figure 45. Message receive rate and distance between UAS and sensor for orbit #3, sensor locaiton #3.

stands of trees). Second, it was unclear if aircraft attitude had any affect on the communication between the UAS and ground sensor. Since the optimal trajectory is nearly at the limit of the ground sensors communication range, it's likely that small variations of the fixed airborne antenna may impact connectivity. Finally, the hardware protocols on communication vary the power output to improve the receive and transmit signal strengths. This power modulation is internal to the radio software and not tracked by telemetry. In other words, the signal strength may be variable, especially once connected. The test team did observe that although the UAS was inside the communication range, it did not connect right away, and as the aircraft flew beyond comm range, it remained connected longer than expected.

Further, the airborne WiFi antenna is mounted on the top of the aircraft (since the UAS belly lands, this is the most logical placement). Although the aircraft is made of foam and with a modest slant angle to the ground sensor, the antenna connects for

most of the duration of the flight. However, the aircraft fuselage may shield the WiFi antenna in certain attitudes. The fuselage also contains the battery packs, autopilot, motor and servos, all of which may contribute to the notion of aircraft shielding. The shape of the antenna pattern may be another suspected reason for connectivity issues. Although the WiFi antenna radiates in a nearly spherical radiation pattern, weaker antenna lobes or null spots exist. The effects of moving WiFi are not well understood as most applications tend to use fixed WiFi antennas.

Unfortunately, these are assumptions with no conclusive evidence. However, based on the team's years of flight test experience and sound engineering judgement, they certainly fall within the realm of possibility.

It appears the ideal communications range is 1.15 km, but in an operational setting, a more conservative, 1 km communication range better accounts for anomalies and other variations.

5.10 Recommendations

First, in order to better improve the methods of calculating the optimal trajectory, several recommendations are made.

1. Improving the communications model is necessary for more consistent results. Although modeling the anomalies is not desired, including stochastic methods may help to predict them.
2. Wind effects should not be discounted. Since the UAS flies relatively slowly, wind can easily effect the ground speed and skew the data collected. Therefore, consideration should be given to flying the optimal trajectory in both directions to determine wind effects on data collection and improve optimization routine to consider these effects. This is also similar to the first recommendation for future flight testing below.

As a result of the flight testing at Wilmington Air Park, a list of recommendations is provided for future testing.

1. Fly initial orbits of the WSN in a TSP like trajectory, collecting wind and sensor data. Then calculate the optimal trajectory based on the known wind conditions and the data required from each sensor. Consider TSP orbits in both directions to account for head and tail winds and their effects on ground speed.
2. During flight test consider the use of a spectrum analyzer to determine radio frequency activity in the area.
3. Through a design of experiments, consider various antenna locations on the UAS. Furthermore, consider a flight test profile with various altitudes, speeds, and aircraft attitudes and compare any data loss with aircraft positioning.
4. Continued characterization of the communications is necessary. Inconsistent connections should be mitigated as much as possible. More robust hardware or bench level testing prior to field use may be required as well. If possible test in an anechoic chamber.
5. The test is limited in scope due to airspace and test range restrictions. A more geographically dispersed and geometrically complex WSN array is recommended.
6. The WiFi antennas are employed beyond their design limits and another radio communications medium is required. Especially considering the various locations the UAS and WSN can be tested, a more robust form of communication is required.

5.11 Conclusion

In conclusion, flight testing validated the optimal trajectory's ability to collect the required data from a WSN. Flight testing the optimal trajectory proved that direct overflight of sensors, as performed in a traditional TSP fashion, can be improved upon, namely decreasing flight time and distance. Although the minimum data was collected from only three of four sensors at the 1.15 km assumed communications range, a more conservative 1 km range is recommended. The 1 km range provides sufficient buffer to handle communications anomalies and increase the likelihood of collecting the minimum data required. Furthermore, the optimal trajectory saved over 32% in flight time and distance over the calculated TSP trajectory. This chapter presented sufficient details of the experiment, results, analysis, and recommendations to warrant further investigation of the theory and methods of this dissertation. A robust analysis of the flight test data and hardware is required, but is beyond the scope of this work.

VI. Conclusions and Recommendations

6.1 Conclusions

This final chapter summarizes the dissertation and presents conclusions, contributions, and recommendations for future research. The work presented successfully developed methods that (1) initially and optimally assign sensors of a WSN to UAS, (2) design optimal flight trajectories for UAS collecting data from their assigned sensors, and (3) re-assign sensors to UAS if they are identified as sensors with relatively higher activity. Specifically, the methodology developed and demonstrated achieved these goals:

1. **Given a WSN and multiple UAS, develop an optimal sensor to UAS assignment algorithm.** The work herein demonstrated that using a simple Euclidean distance measure, the k -means++ clustering algorithm, optimized the distance of a sensor to a given centroid. The number of available UAS equalled the number of centroids (and clusters) that would be created. Once assigned to a centroid, a sensor is assigned to a cluster and ultimately to a UAS.
2. **Given a WSN and UAS, develop an optimal control method to ensure sufficient data collection from all sensors, in the least amount of time and with minimal control effort.** The work presented a continuous time optimal control problem formulated and solved for the optimal trajectory in an assigned sensor cluster. The optimal control problem minimized the time and control effort required to collect a specified minimum amount of sensor data. This problem formulation was a modified traveling salesman problem with neighborhoods and the control policy enables the airborne vehicle to change altitude, heading and velocity to best communicate with each ground sensor

via omni-directional antennas. The optimal trajectory is solved for by direct orthogonal collocation methods also known as a pseudospectral method. Each unmanned aircraft is then tasked to fly the optimal trajectory within its assigned cluster, survey, and collect the minimum amount of required data from each assigned sensor.

3. **Conditioned on the data collected as the problem evolves, develop a method to increase the visitation rates of the highest activity sensors and decrease or maintain the visitation of the least active sensors.** The work herein presented a method to increase revisit rates at sensors identified as high activity. Gaussians were implemented to represent the clusters since they were better suited to handle the various scenarios than k -means++. As sensors were weighted due to their high activity, the use of an Expectation Maximization method, on page 68, resulted in Gaussians reshaped to better segregate high activity sensors from relatively lower activity sensors. UAS are then assigned to new clusters and re-tasked with updated optimal trajectories. The new assignments and optimal flight trajectories are calculated to ensure increased revisit rates over the high activity sensors, while maintaining or decreasing time over sensors with relatively less activity.

Collectively, this research demonstrates that methods are possible to optimally assign WSN sensors to UAS and design optimal trajectories to collect data from assigned sensors. In simulation and flight test, optimal trajectories were shown to save between 14% and 32% flight time and distance when compared to traditional traveling salesmen solutions, while still collecting the minimum specified data.

6.2 Contributions

This dissertation made several specific contributions in the area of data collection from WSN using a UAS.

1. When multiple UAS are available, an initial sensor to UAS assignment is required. The k -means++ with Euclidian distance metric proved to be an ideal method.
2. For subsequent sensor to UAS assignments, various locations and priority given to high interest sensors demanded a different approach to the assignment methodology. Seeded with the k -means++ solution, a multi-variate Gaussian mixture models with Expectation Maximization can be used to assign sensors to UAS.
3. The formulation of an optimal control problem to solve the data collection problem which incorporated a communications model as a constraint. The solution was then found by direct orthogonal collocation methods. Additionally, a initial guess methodology was is provided as a means to overcome issues with little gradient information.
4. An simple and flexible communications model. If the behavior of the communications are known, a sigmoid based empirical representation can be made. With adjustments to stiffness, maximum data rate, or maximum communications distance, the model proves to be a sound representation of communications.
5. An incremental procedure to convert a numerical solution in MATLAB to a flyable set of waypoints for flight test. Also from flight test, an improved understanding of the communications environment and UAS data collection scheme.

6.3 Potential Future Research

The following areas are the major areas identified for future research.

1. Finding an adequate cost function weighting algorithm proved to be elusive. Future research should include development of a system or sound methodology to demonstrate a weighting scheme that automatically updates based on the problem conditions. Alternatively, a parameter optimization may identify a set of weights applicable to a wider range of problems. Although an adequate method was used for the problems presented, a more mathematically rigorous method is still desirable.
2. The communications model is perhaps the most critical modeling component. Although the optimal control problem was agnostic to the type of communication model, its output remains a constraint on the overall problem. Future research must consider a communications model that accurately represents the relationship between the UAS and sensor. Along with the velocity and heading rate limits, the communications model determined the optimized distance necessary to maximize the communications rate. This Euclidean distance to the sensor ultimately determined the shape of the optimal trajectory. Future research may also consider alternates to data rate, such as packets received or signal strength instead.
3. Constraining the final heading to 2π limits the UAS from performing any loiter maneuver around individual sensors. Exploring a method to vary this final heading condition is recommended. In turn, this enables the UAS to perform a loiter around particular sensors for increased data collection in addition to other control options such as reduced velocity, changing altitude, and altering the flight path. Determining if additional time, and possibly a loiter maneuver

at any one sensor is conditioned on the amount of data required from individual sensor, a value not known until the sensor is visited.

4. Methodically weighting the high interest sensors in re-assignment method may also prove to be a valuable area for future research. The incremental method currently used eventually led to the correct weights. However, future methods may immediately pick the correct weight, without incrementally searching for them, based on the sensors selected.
5. Consider a secondary analysis of the optimal trajectory in NASA's Optimal Trajectory by Implicit Simulation program. Variations on atmospheric and or higher degrees of freedom aircraft models can help to further test effectiveness of calculated trajectories.
6. Variations in some optimal trajectory parameters led to large changes in solutions. Many parameters effect the outcome of the optimal trajectory routine and a thorough sensitivity analysis of the optimality terms is warranted. In tern, this can lead to improvements that may lead to increased robustness of the overall solution.
7. A real time implementation of the end to end scenario with on-board UAS processing. With multiple UAS and a large enough flight test range, consideration should be given to exercising the entire scenario. Furthermore, packaging of the assignment and trajectory optimization routines for on-board processing will only underscore the effectiveness of this method.

6.4 Summary

In summary, the sensor to UAS assignment and optimal trajectory methods proved to successfully collect data from a WSN, enabling a persistent surveillance over a

region of interest. Flight testing verified the UAS the communications model. Furthermore, the ability to concentrate on areas within the WSN gives the methodology more flexibility and applicability. Specifically, these techniques solve the patrol and isolation in a WSN problem introduced by AFRL as developed in Chapter I. These techniques also contribute to the DoD's vision of determining optimal solutions for unforeseen situations by unmanned systems. The methods developed are novel, yet simple enough to be implemented by any UAS and WSN users. Higher fidelity UAS models and improved communications models will only benefit these processes and help to make them more robust.

Appendices

Each of the following three Appendices, A, B, and C, contain a conference paper published as a part of this research effort. The three papers represent an evolutionary process where the second and third papers updated and improved upon the first, and where this dissertation eventually improved upon the second and third papers. For this reason, the work in Appendix A has been superseded. Namely updates to the cost function, heading states, and scenarios make the first paper obsolete. The three conference papers are:

1. Appendix A - Optimal Airborne Trajectories for Data Collection from Emplaced Ground Sensor Arrays, January 2014.
2. Appendix B - Optimal Airborne Trajectories for Data Collection from Wireless Sensor Networks by Direct Collocation Methods, January 2015.
3. Appendix C - Autonomous Flight Path Planning for Traffic Monitoring in Wireless Sensor Networks, January 2015.

Appendix A. Conference Paper #1 (January 2014)

This appendix contains the first conference paper published as part of this research effort. It was published and presented at the AIAA 2014 SciTech Guidance, Navigation, and Control Conference in Washington D.C., in January 2014 [28]. This paper is superseded by the conference papers published in January 2015, found in Appendices B and C.

Optimal Airborne Trajectories for Data Collection from Emplaced Ground Sensor Arrays*

Nidal M. Jodeh[†], Timothy E. Coon[‡], Tadeusz J. Masternak[§],
Richard G. Cobb[¶], and Jeremy S. Agte^{||}
Air Force Institute of Technology, Wright-Patterson AFB, OH, 45433

The optimal flight trajectory of an unmanned aerial vehicle acting as a primary central communication node for multiple randomly placed unattended ground sensors in minimum time is considered. An algorithm is developed to ensure minimum required data transfer between an airborne agent and a sparse unattended ground sensor network. In this modified travelling salesman problem, the control policy enables the airborne vehicle to change altitude, heading and velocity to best communicate with each ground sensor via an on-board omni-directional antenna. Optimal flight trajectory planning algorithms are proposed to ensure the unmanned aerial vehicle successfully receives the required data transmission from each ground sensor in minimum time.

Nomenclature

| | |
|----------------------|---|
| x | Position along the x -axis (m) |
| y | Position along the y -axis (m) |
| z | Position along the z -axis (m) |
| \dot{x} | Velocity in x direction (m/s) |
| \dot{y} | Velocity in y directions (m/s) |
| \dot{z} | Velocity in z direction (m/s) |
| v | Airspeed of the UAV (m/s) |
| θ | Heading angle with respect to inertial frame (rad) |
| ϕ | Pitch angle with respect to inertial frame (rad) |
| $\ddot{}$ | Acceleration of the UAV (m/s^2) |
| $\dot{\theta}$ | Heading angle time rate of change (rad/s) |
| $\dot{\phi}$ | Pitch angle time rate of change (rad/s) |
| C_j | Channel capacity of communication with sensor j (bps) |
| B | Channel bandwidth (Hz) |
| N | Noise floor of the channel (W) |
| S_j | Average received power from sensor j (W) |
| $\frac{S}{N}$ | Signal to noise ratio |
| P | Equivalent isotropic radiated power of each UGS (W) |
| L_j | Free-space spreading loss for sensor j |
| d_j | Distance between UAV and sensor j (m) |
| f | Wavelength of the carrier frequency (m) |
| c | Speed of light (m/s) |
| DP | Minimum Data Package Size ($bits$) |
| s | Number of sensors |
| J | Cost functional |
| t | Time (s) |
| α | Convex combination coefficient |
| \mathbf{u} | Control vector |

*Distribution A: Cleared for public release, case number 88ABW-2013-5148.

[†]PhD Student, Department of Aeronautics and Astronautics, 2950 Hobson Way, Lt Col, USAF, Senior Member AIAA.

[‡]PhD Student, Department of Aeronautics and Astronautics, 2950 Hobson Way.

[§]PhD Candidate, Department of Systems Engineering and Management, 2950 Hobson Way, Senior Member AIAA

[¶]Associate Professor, Department of Aeronautics and Astronautics, 2950 Hobson Way, Associate Fellow AIAA

^{||}Assistant Professor, Department of Aeronautics and Astronautics, 2950 Hobson Way, Lt Col, USAF, Senior Member AIAA

U Set of all admissible optimal control vectors

Subscript

j Sensor number ($j=1,2,3,\dots,s$)

o Initial

f Final

I. Introduction

The optimal control of a UAV to collect data from a randomly placed ground sensor network is considered. A sparsely populated unattended ground sensor (UGS) network can be utilized to protect vital assets or remote positions, such as Forward Operating Bases. Unattended ground sensors collect, record, and report environmental disturbances within their immediate vicinity. Disturbances may be recorded as Doppler, seismic, magnetic and infrared information, and reported to overflying UAVs with the necessary time information¹. For example, intruders or inconspicuous passing vehicles can be recorded as seismic disturbances, time-stamped, and transmitted to the overflying UAV. Unattended ground sensors are assumed to be low-power sensors and cannot communicate with other UGSs or satellites. Commonly, UGSs rely on a fixed, ground-based central collection node, in proximity to the UGS, to collect and transmit vital information. This central node requires line of sight with UGSs, communication link with operators, power, and physical protection. This central node is a single point of failure for local UGS networks and can prove difficult to place and maintain. It is advantageous to deploy UGSs over a larger geographical area and this is easier if they can be made independent of a central collection node. The UAV is ideally suited to collect data from an UGS network over a vast area, beyond the communications range of a ground based central collection node. However, employing a UAV has trade-offs. These include weather constraints, airspace and communication frequency management requirements, UAV logistics and maintenance, as well as processing and response time.¹

II. Motivation

The scenario investigated herein is based on research by Kingston¹, Marshall et al.², and Klesh et al.³ and is of interest to the Air Force Research Laboratory (AFRL). AFRL proposes a concept replacing the central node with a UAV, where the UAV visits all the UGSs to collect and relay sensor data to necessary users, obviating the ground-based central collection node. This architecture enables one UAV to serve a large number of geographically distributed UGSs as opposed to a single central node addressing UGSs with a specified communications range. AFRL's scenario is divided into four phases; patrol, isolation, delivery, and response. During the patrol phase the UAV is tasked to visit the UGSs in a deliberate search pattern as driven by areas of high interest, airspace limitations, and likely areas of intrusion. The isolation phase is triggered by a UGS detection of an intrusion. Given this detection, the UAV is tasked to autonomously decide on the next likely location of detection and alter its flight path. The goal of the isolation phase is to capture the intruder on an electro-optical sensor. Once the intruder has been imaged by the UAV, the delivery phase ensures the appropriate agencies are notified. Finally, the response phase is triggered when the image is analyzed and the proper course of action is taken. This paper proposes a possible optimal control based solution to the patrol phase of the scenario.

III. Problem Statement

The objective is to minimize the required UAV flight time to collect the minimum required data from a sparsely populated, randomly placed UGS network. This problem can be viewed as a modified traveling salesman problem.¹ An omni-directional antenna is used to collect data from the low-power UGSs. The UAV is permitted to vary heading, altitude, and airspeed.

The UAV antenna is required to be in contact with each UGS long enough to ensure transmission of a fixed data package size. For the purposes of the problem formulation, the UAV starts and ends its flight from the same airfield, set as the initial and final boundary conditions on the UAV's trajectory.

Model and Assumptions

Components are modeled along with several assumptions to simplify the problem. For the UGSs, their transmission is continuous and isotropic over the entire field in consideration with a fixed transmitted power. UGS locations are

known to the UAV a priori and do not move once placed. The UAV is modeled as a point mass system and external forces are not considered. Stochastic disturbance inputs such as wind and other noise are not considered. The mission requires the UAV to take off and land at the same airfield in same direction for each data collection orbit. Only one orbit of the UGS network is calculated. Additional orbits are presumed to be multiples of the single orbit and the number of additional orbits is subject to user requirements, UAV limitations, etc. The UAV airfield is typically located approximately at the geographic center of the UGS network, typical for forward operations base protection. It is assumed the UAV modeled in this paper is a small Tier N/A or Tier 1 type vehicle, as defined by the US Air Force.⁴ Many of these vehicles can reach their max speeds in a negligible amount of time. The initial and final velocity states are on the runway and set to zero at the same location. Although the UAV takes off and lands from the same location, the minimum airspeed is set to the stall speed. In other words, the take off and landing are assumed negligible, and do not influence the optimal control problem. Finally, the UAV and each UGS is equipped with a single omni-directional antenna that is assumed homogeneous and attenuated only by free space spreading loss. No geographic limitations or azimuth and elevation requirements were imposed between the UAV and the UGSs. Two-way communication and error checking are not considered.

IV. The Optimal Trajectory Problem

In this section, a three dimensional mathematical model is developed for the optimal flight path problem. The dynamic model is described by the equations of motion subject to certain constraints. The velocity, heading, and pitch angles are three independent controls. The goal is to minimize the time required to collect a minimum amount of required data from each of the UGSs in a given sensor network. An airborne agent, the UAV, is expected to fly an optimal trajectory to meet the stated goals. The problem is developed in continuous time.

A. Equations of Motion

The problem is defined by six states. The first three states are the location of the UAV in the $x(t)$, $y(t)$, and $z(t)$ directions with respect to an origin in the inertial frame. The next three states are the airspeed of the UAV, $v(t)$, the heading angle, $\theta(t)$, measured positive counterclockwise in the xy -plane from the inertial x -axis, and the pitch angle, $\phi(t)$, measured positive from the inertial xy -plane. The controls are the acceleration of the UAV, $\dot{v}(t)$, the heading angle rate in the inertial frame, $\dot{\theta}(t)$, the pitch angle rate in the inertial frame, $\dot{\phi}(t)$. All states and controls are time-dependent and the variable t will be dropped for the balance of the paper. The problem is bounded by limits on all six states and all three controls.

The equations of motion are defined by the relationship between the states and the controls. They are derived from the geometry and differential kinematics and are given in Equation (1). A geometric definition of the variables is supplied in Figure 1 where the UAV, represented by a point mass, resides at the origin of the coordinate system.

$$\dot{x} = v \cos \phi \cos \theta \quad (1a)$$

$$\dot{y} = v \cos \phi \sin \theta \quad (1b)$$

$$\dot{z} = v \sin \phi \quad (1c)$$

Communications Link

The objective is to obtain the data stored on the sensors in a minimum amount of time via radio communications. The data rate of transfer between each UGS and the UAV is the driving dynamic relationship. The data rate for the communications bandwidth is estimated to be the channel capacity as established by Shannon's channel capacity, Equation (2), the data rate for which the probability of bit error becomes arbitrarily small.^{5,6} By implementing the channel capacity expression, it is assumed the transmissions having any bit errors result in zero significant data collected while all other transmissions are received and decoded with no error or other limitations on the data rate.

$$C_j = B \log_2 \left(1 + \frac{S_j}{N} \right) \quad (2)$$

The values of both the channel bandwidth, B , and the noise floor (average interference noise), N , are assumed to be constant. For this paper, two noise floor values are used in various test cases. Therefore, the channel capacity is a function of only the average received power, S_j . The power received is determined at each state by the application of the link-power budget equation.⁵ For the problem presented, both the transmitting and receiving antennas are assumed

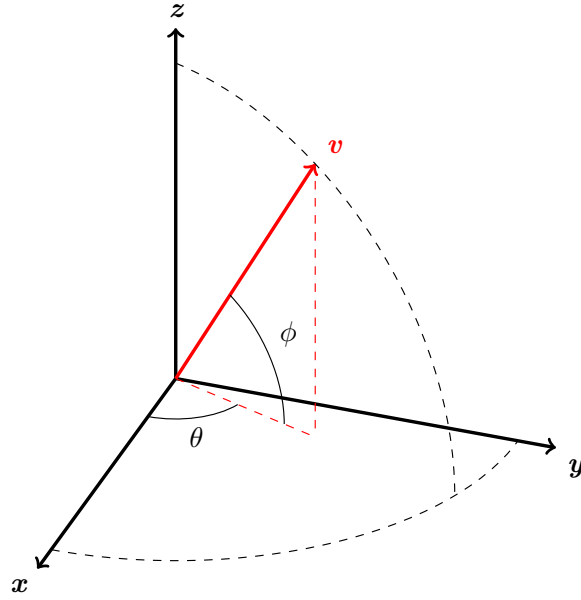


Figure 1. State variables in relation to the inertial frame coordinate axes.

ideal omni-directional antennas and there is unity gain associated with each. The only loss considered is the free-space spreading loss as it is solely a function of the distance between the UGS transmitter and the UAV receiver. Given these assumptions, the expression for received signal power is given in Equation (3) where the free-space spreading loss is determined with Equation (4). The expression for the channel capacity of the communication with sensor j is now written explicitly as a function of the separation distance, d_j , in Equation (5).

$$S_j = \frac{P}{L_j} \quad (3)$$

$$L_j = \left(\frac{4\pi d_j f}{c} \right)^2 \quad (4)$$

$$C_j = B \log_2 \left(1 + \frac{P}{N \left(\frac{4\pi d_j f}{c} \right)^2} \right) \quad (5)$$

B , P , and N are assumed the same for each sensor, j . Consider this equation in the context of the problem in which the UAV traverses the flat landscape in which s UGSs are placed with significant separation. As the UAV changes position, the distance between the UAV and the fixed UGSs also changes as does the channel capacity. Because the communication is intended to collect information stored in the UGS's memory, the parameter of interest is the volume of data transferred, measured as the number of bits. The volume of data received over the time interval is then the integral sum of the channel capacity, Equation (6). It is assumed the data transmitted meets a minimum fixed data package size, DP , required for every sensor. It is also assumed the UAV continuously receives data regardless of the signal strength, where, in reality, a minimum threshold on the signal power severs the link altogether. This discrepancy is negligible at great distances as the data rate becomes inconsequential, however the influence of a decaying data rate on all problem scenarios is well understood.

$$DP_j \leq \int_{t_0}^{t_f} C_j dt \quad \forall j \quad (6)$$

To relate the calculation of the data package size to the problem formulation, the distance to sensor j is expressed in terms of the state variables and the j^{th} UGS's position coordinates, Equation (7).

$$d_j = \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad (7)$$

With the system dynamics and the integral path constraint, the formal definition of the continuous optimization problem is complete. The solution is the control which minimizes the cost function, Equation (8), subject to the system dynamic, Equation (1), the integral path constraints, Equation (6), and the boundary conditions, Equation (11).

The optimal control problem is now formulated to find the control, \mathbf{u} , in the set of admissible controls, \mathbf{U} , that minimizes the following cost functional

$$\min_{\mathbf{u} \in \mathbf{U}} J = \alpha t_f + (1 - \alpha) \int_{t_0}^{t_f} (\dot{v}^2 + \dot{\theta}^2 + \dot{\phi}^2) dt \quad (8)$$

subject to the dynamic constraints

$$\dot{x} = v \cos \phi \cos \theta \quad (9a)$$

$$\dot{y} = v \cos \phi \sin \theta \quad (9b)$$

$$\dot{z} = v \sin \phi \quad (9c)$$

where the controls are

$$\mathbf{u} = (\dot{v}, \dot{\theta}, \dot{\phi}) \quad (10)$$

with the boundary conditions,

$$(x(t_0), y(t_0), z(t_0), v(t_0), \theta(t_0), \phi(t_0)) = (x_0, y_0, z_0, v_0, \theta_0, \phi_0) \quad (11a)$$

$$(x(t_f), y(t_f), z(t_f), v(t_f), \theta(t_f), \phi(t_f)) = (x_f, y_f, z_f, v_f, \theta_f, \phi_f) \quad (11b)$$

the path constraints,

$$x_{min} \leq x \leq x_{max} \quad (12a)$$

$$y_{min} \leq y \leq y_{max} \quad (12b)$$

$$z_{min} \leq z \leq z_{max} \quad (12c)$$

$$v_{min} \leq v \leq v_{max} \quad (12d)$$

$$\theta_{min} \leq \theta \leq \theta_{max} \quad (12e)$$

$$\phi_{min} \leq \phi \leq \phi_{max} \quad (12f)$$

$$\dot{v}_{min} \leq \dot{v} \leq \dot{v}_{max} \quad (12g)$$

$$\dot{\theta}_{min} \leq \dot{\theta} \leq \dot{\theta}_{max} \quad (12h)$$

$$\dot{\phi}_{min} \leq \dot{\phi} \leq \dot{\phi}_{max} \quad (12i)$$

$$(12j)$$

and the integral constraints

$$\int_{t_0}^{t_f} C_j dt \geq DP_j \quad \forall j. \quad (13)$$

V. Solution

A. Problem Specific Values

This section specifies the problem specific values. The values are derived from various sources and consistent with nominal USAF Tier 1 UAVs and commercial small UAVs in use today.^{4**} Specifications selected for the UGS model are based on commercially available systems^{††} and also based on discussions with AFRL personnel currently employing UGS systems.^{‡‡} These values are approximations and do not represent a specific UGS. UGS and UAV values are collected in Table 1 and Table 2 respectively.

The weighting factor on the final time and controls, α , is set to 0.5, equally weighting the terms of the cost functional. The value of α is determined by trial and error and is user specified.

^{**}Reference to the MLB Corp SuperBAT and BAT 4 UAVs for example, <http://spyplanes.com/products-s-bat/>

^{††}Reference to the UGS Dragon Sensor for example, <http://beyondsof.com/wp-content/uploads/2012/02/UGS-Dragon-Sensor-datasheet.pdf>

^{‡‡}Email correspondence with Dr Dave Gross, ICE-T Principle Investigator, Wright State Research Institute and Dr Derek Kingston, AFRL

Table 1. Communications Link Values and Settings

| Parameter | Value |
|-----------|---------------------------|
| B | $1(Hz)$ |
| N | 10^{-3} to $10^{-2}(W)$ |
| P | $10^{-2}(W)$ |
| DP | 3000 (bits) |

Table 2. UAV Limits

| Parameter and Limits |
|--|
| $x(t_0), y(t_0), z(t_0), v(t_0), \theta(t_0), \phi(t_0) = 0$ |
| $x(t_f), y(t_f), z(t_f), v(t_f), \theta(t_f), \phi(t_f) = 0$ |
| $-10,000 \leq x \leq 10,000 (m)$ |
| $-10,000 \leq y \leq 10,000 (m)$ |
| $0 \leq z \leq 10,000 (m)$ |
| $20 \leq v \leq 50 (m/s)$ |
| $-180 \leq \theta \leq 180 (deg)$ |
| $-12 \leq \phi \leq 12 (deg)$ |
| $-1 \leq \dot{v} \leq 1 (m/s^2)$ |
| $-30 \leq \dot{\theta} \leq 30 (deg/sec)$ |
| $-6 \leq \dot{\phi} \leq 6 (deg/sec)$ |

B. Optimal Trajectory Test Case Set Up

Optimal trajectories are found for both two- and three-dimensional test cases. The two-dimensional cases are used primarily as baseline checks of the optimization routine, where the solution is easily interpreted. For the two-dimensional cases, the UAV and UGSs are fixed at the same altitude. In the three-dimensional cases, UGS are placed at various x, y, and z coordinate locations, simulating varying terrain elevations. Optimal flight paths were solved for 1, 4, 8 and 25 sensors. For separate test cases, the noise floor is increased from $10^{-3}(W)$ to $10^{-2}(W)$, enabling a comparison of the same sensor field in different signal noise environments. The minimum data package required from each UGS for all the test cases is 3000 bits and is user- and hardware-dependent based on the data required. There are numerous combinations of data package size, UGS count, UGS locations, and other variables, and testing all the possible cases is unnecessary. The test cases are outlined in Table 3 and the optimal solutions are presented and discussed in section VI.

C. Solution Method

The MATLAB-based, open source, General Purpose Pseudospectral Optimal Control Software-II (GPOPS-II) is the program used to solve for the optimal trajectories. GPOPS-II directly solves the continuous time optimal control problem.⁷ GPOPS transcribes a continuous-time optimal control problem into a discrete finite-dimensional static optimization problem that is solved as a Non-Linear Programming (NLP) problem. GPOPS-II is implemented using the *hp*-adaptive version of the Radau pseudospectral method. This particular method is an orthogonal collocation Gaussian quadrature implicit integration method in which the collocation is performed at the Legendre-Gauss-Radau quadrature points.⁸ An adaptive mesh refinement method is implemented that determines the number of mesh intervals and the degree of the approximating polynomial for each mesh interval in order to satisfy the specified error tolerance.⁸ GPOPS-II can be set to prioritize finding the degree of the approximating polynomial before refining the mesh interval or vice versa.

Table 3. Two Dimensional UAV-UGS Optimal Trajectory Test Cases

| Case No. | No. of UGSs | UGS Coordinates | | | Noise Floor (W) |
|----------|-------------|-----------------|------------|------------|--|
| | | x (m) | y (m) | z (m) | |
| 1 and 2 | 1 | 0 | 10^3 | 0 | <i>for Case 1: 10^{-3} for Case 2: 10^{-2}</i> |
| 3 and 4 | 4 | -10^3 | 0 | 0 | <i>for Case 3: 10^{-3} for Case 4: 10^{-2}</i> |
| | | 0 | 10^3 | 0 | |
| | | 10^3 | 0 | 0 | |
| | | 0 | -10^3 | 0 | |
| 5 and 6 | 8 | -10^3 | 0 | 0 | <i>for Case 5: 10^{-3} for Case 6: 10^{-2}</i> |
| | | -10^3 | -10^3 | 0 | |
| | | 0 | -10^3 | 0 | |
| | | 10^3 | -10^3 | 0 | |
| | | 10^3 | 0 | 0 | |
| | | 10^3 | 10^3 | 0 | |
| | | 0 | 10^3 | 0 | |
| | | 10^3 | 10^3 | 0 | |

D. GPOPS-II Settings

For all the test cases, GPOPS-II is set to use the IPOPT NLP solver. The derivative approximation used is “sparseCD” and second derivatives are provided. The mesh method is set to “hp1,” where GPOPS-II increases the degree of the approximating polynomial up to a maximum allowable degree prior to dividing the interval into sub-intervals.⁹ The error tolerance is set to 10^{-3} . “RPMintegration” is enabled and manual scaling is performed. All results presented are from computations performed using MATLAB 2012b on a 2012 MacBook Pro with a 2.9 GHz Intel Core i7 processor and 8 GB of 1600 MHz DDR3 memory.

VI. Results

The following results are determined for each test case of Table 3. Optimal solutions are presented for all test cases and all meet the mesh error tolerance. Although GPOPS-II is allowed to run up to 25 mesh iterations, most solutions converged in one to three mesh iterations. The results are presented to better illustrate to the reader the optimal solution in a layered method. Two-dimensional cases for one, four and eight sensors are displayed. Three-dimensional cases are reported for 8 and 25 sensors. Table 4 details the combination of UGSs, Noise floor, and final time for the two-dimensional cases.

Table 4. Two-Dimensional Optimal Trajectory Test Case Results, Figures 2-5.

| Case No. | No. of UGSs | Noise Floor (W) | Time Final (sec) | Figure |
|----------|-------------|--------------------|---------------------|-------------|
| 1 and 2 | 1 | Case 1: 10^{-3} | 375 | 2(a) |
| | | Case 2: 10^{-2} | 434 | 2(b) |
| 3 and 4 | 4 | Case 3: 10^{-3} | 410 | 3(a) |
| | | Case 4: 10^{-2} | 897 | 3(b) & 4(a) |
| 5 and 6 | 8 | Case 5: 10^{-3} | 545 | 5(a) |
| | | Case 6: 10^{-2} | 1196 | 5(b) |

Figure 2 illustrates cases 1 and 2 for two-dimensional optimal trajectory solutions for a single UGS. The UAV successfully receives the minimum data required in 375 seconds in the low noise environment and the optimal trajectory is shown in Figure 2(a). In Figure 2(b), the noise floor is raised and the time increases to 434 seconds to acquire the minimum data required. In order to satisfy the 3000-bit minimum data requirement, the distance between the sensor and the UAV over the optimal trajectory is increased as expected from Equation (13). These initial results are logical and used as a benchmark for successive builds of the following test cases.

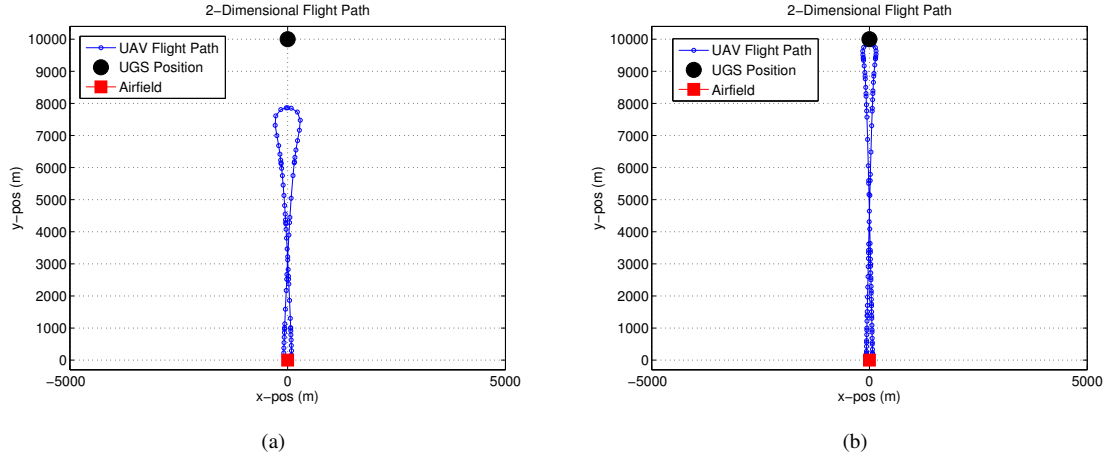


Figure 2. Optimal UAV trajectory for a single UGS with varied noise floors. (a) Case 1: One UGS; Range = 10^3 m; Noise Floor = 10^{-3} W. (b) Case 2: One UGS; Range = 10^3 m; Noise Floor = 10^{-2} W.

Figure 3 illustrates cases 3 and 4, two-dimensional optimal trajectory solutions for a four UGS network. Again, these cases are used as tests to reinforce and test the problem formulation and calculation by GPOPS-II. The optimal trajectory for two noise floor conditions, 10^{-3} W and 10^{-2} W, is seen in Figure 3(a) and 3(b). Also provided, in Figures 4(b) and 4(c), are plots of the states of case 4. Figure 4(c) shows smooth changes of the heading, pitch, and velocity states, typical of actual aircraft. The plot of Figure 4(d) displays the data transfer rate of each UGS versus time over the duration of the flight. The data bolsters the optimal solution because the data rate of UGS j increases as the distance between the UAV and UGS j (shown in Figure 3(b)) decreases. Figure 5 illustrates a two-dimensional optimal trajectory solution for an eight sensor network.

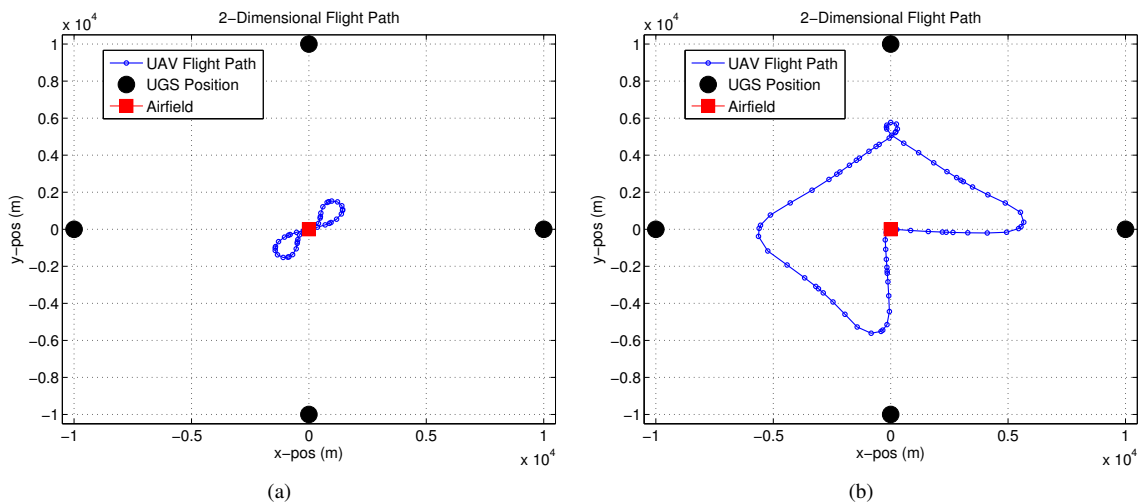


Figure 3. Optimal UAV trajectory for an four UGS array with varied noise floors. (a) Case 3: Four UGSs; Range = 10^3 m; Noise Floor = 10^{-3} W. (b) Case 4: Four UGSs; Range = 10^3 m; Noise Floor = 10^{-2} W.

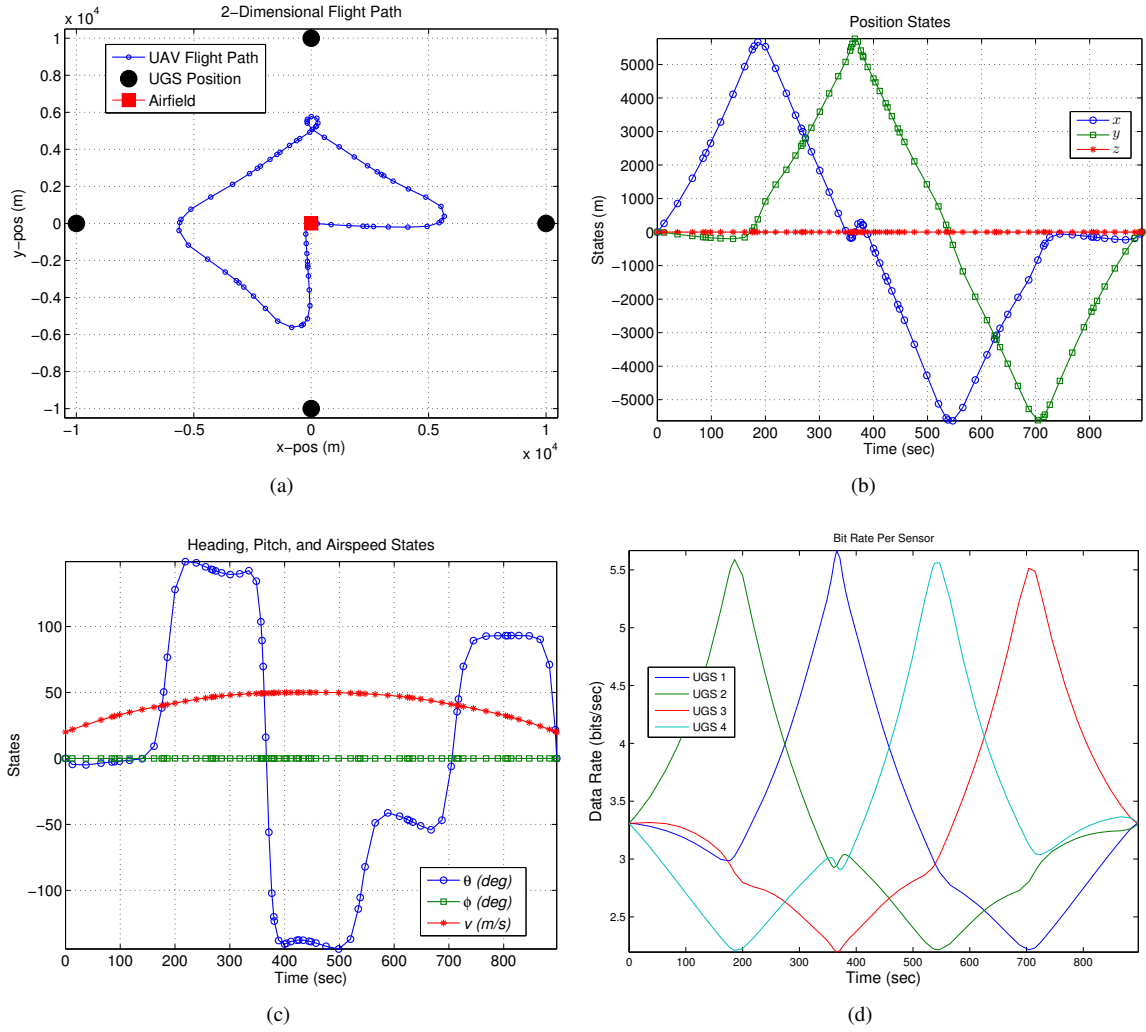


Figure 4. Case 4: (a) Optimal UAV trajectory for a two-dimensional array of four UGSs in a high-noise environment. (b) and (c) The states over the flight duration. (d) The data rate of the transmission from each sensor with respect to time.

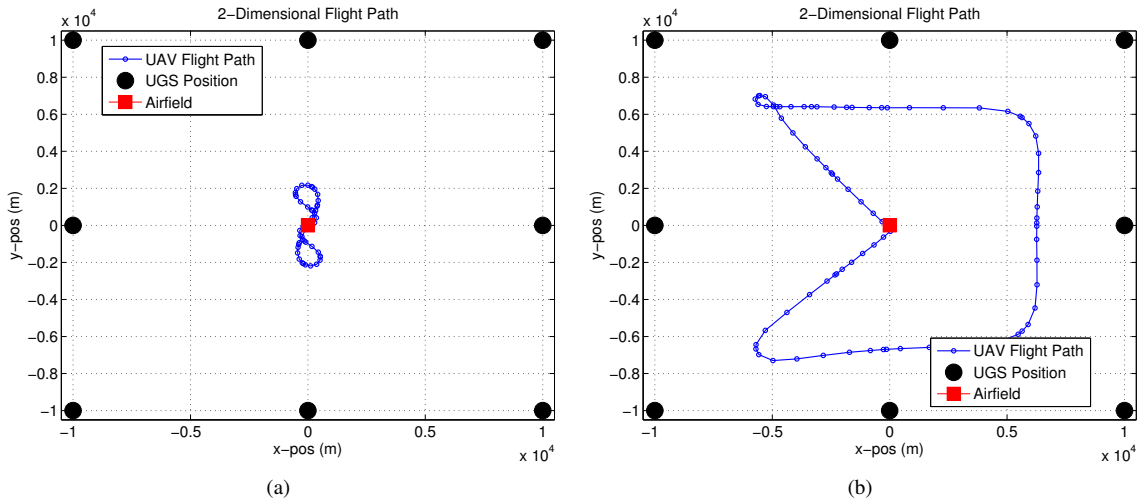


Figure 5. Optimal UAV trajectory for an eight UGS array with varied noise floors. (a) Case 5: Eight UGSs; Range = 10^3 m; Noise Floor = 10^{-3} W. (b) Case 6: Eight UGSs; Range = 10^3 m; Noise Floor = 10^{-2} W.

Figure 6(a) illustrates eight randomly placed UGSs in three-dimensions from a top view. The UGSs are placed at various elevations to simulate varying terrain. The elevations are exaggerated for the simulation to better illustrate the optimal trajectory adapting to the sensor's elevated position. Figure 7 illustrates the same scenario but with 25 UGSs.

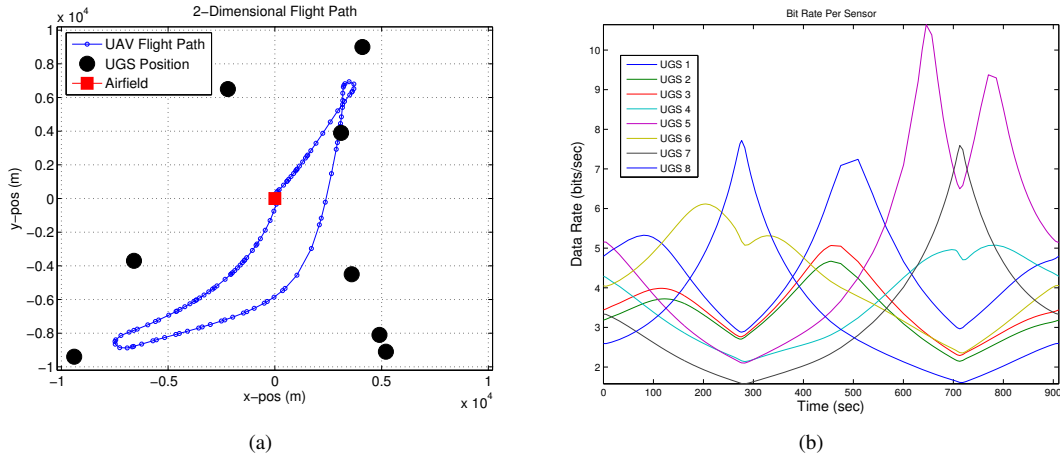


Figure 6. Optimal UAV trajectory for eight, randomly placed UGSs at various elevations in a high noise environment. (a) Top view of flight path, (b) Data rates

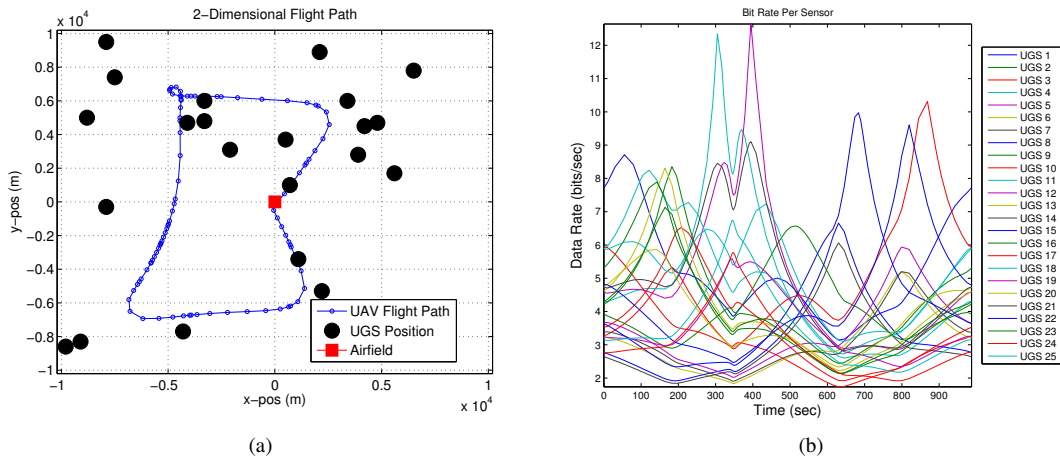


Figure 7. Optimal UAV trajectory for 25, randomly placed UGSs at various elevations in a high noise environment. (a) Top view of flight path, (b) Data rates

VII. Discussion

The optimal flight path solutions are not necessarily intuitive. In the four-UGS case for example, if planned by a human, one may expect to take off from the airfield at the center of the plot and fly a circle with fixed radius about equal to the four sensors and land. However, the optimal trajectory is noticeably different. Based on Figure 4(d), the initial data rates at time zero are considered the data rates received at the UAV while still on the airfield. Based on Figure 4(d), given the data rate received at time zero is 3.3 bits/sec and the minimum data required is 3000 bits, the UAV would then require 909 seconds to collect all the necessary data if it never left the airfield. By contrast, flying the optimal trajectory required 897 seconds, an improvement of only 12 seconds. Although this gain is not significant when using four sensors, the gains are more noticeable when using an increased number of sensors. Figure 6(b) illustrates the data rates of eight randomly placed sensors at various elevations. The furthest sensor, UGS number 8, has the lowest data rate at time zero, 2.6 bits/sec. If the UAV never leaves the airfield, it would require 1,154 seconds

to collect 3000 data bits from UGS number 8. However, flying the optimal trajectory resulted in a total flight time of 910 seconds, a savings of 244 seconds, or a 21% improvement. This trend continues for more complex cases with an increased number of UGSs.

The performance specifications of the UAV and UGS can vary dramatically, and are largely hardware dependent. The noise floor is subject to signal noise in the area at the time of the data collection. Although these test cases compare changes in the noise floor for varying numbers of UGSs, many other combinations of equipment specifications and environmental considerations can affect results, and are beyond the scope of this study.

Several challenges were also overcome during the set up and execution. First, the numerical scaling proved to be more critical than expected. The bounds on flight time, velocity, range, and minimum data packet size required appropriate scaling to work within the numerical accuracy of the computer. For example, setting the data packet size upper bound arbitrarily high, say 10^6 , caused long computation time and infeasible solutions. The same was true of setting the noise floor too low. To increase robustness for a wide variety of problems, appropriate scaling is needed.

VIII. Conclusion

An optimal flight path for a UAV acting as the primary gateway node for a network of UGSs is demonstrated. The optimal path and controls satisfy all the constraints and meet the main objective of receiving an entire fixed data package size from each UGS. For a variety of test cases the Gauss Pseudospectral Method is shown to be a feasible and practical method to determine the optimal control for a real-world scenario. The results show the optimal flight path is not as intuitive as simply flying from UGS to UGS.

A flight test of the optimal trajectory is proposed in collaboration with the Air Force Research Laboratory's (AFRL) Control Science Center of Excellence at Wright Patterson Air Force Base in Dayton, Ohio. AFRL's Intelligent Control and Evaluation of Teams (ICE-T) program actively flight tests UAVs with ground sensor networks.² Ongoing flight tests could incorporate a flight test sequence where the ground sensor positions, airspace and terrain information, as well as other relevant communication and UAV specifications are input to GPOPS-II. An optimal control vector is then generated and uploaded to the UAV and UGS data received during a flight test is evaluated to ensure program requirement minimums are met.

Additional future work includes a higher fidelity UAV model, including stochastic inputs such as wind and expanding the mission envelope to include optimal trajectories for multiple orbits with take-off and landing phases. Take-offs and landings will also be considered from within the sensor network, as in this paper, and from bases outside the sensor network.

Acknowledgement

The authors would like to especially recognize and thank Dr Derek Kingston, Air Force Research Labs, Control Science Center of Excellence, who graciously provided us with the challenging problem concept and background information and to Colonel Angela Suplisson for her endless encouragement and support. Cleared for public release, case number 88ABW-2013-5148.

References

- ¹ Kingston, D., "Intruder Tracking Using UAV Teams and Ground Sensor Networks," Tech. rep., Control Science Center of Excellence, Air Force Research Labs, Wright Patterson AFB, OH, USA, 2011.
- ² Marshall, C., Mears, D. M., Kingston, D. D., and Rasmussen, S., "2010 ICE-T Cooperative Control Flight Testing," AIAA, 2011.
- ³ Klesh, A. T., Kabamba, P. T., and Girard, A. R., "Path Planning for Cooperative Time-Optimal Information Collection," *2008 American Control Conference*, 2008.
- ⁴ "Unmanned Systems Integrated Roadmap FY2011-2036," No. ADA558615, Office of the Under Secretary of Defense (Acquisition Technology and Logistics), Washington DC, Oct 2011.
- ⁵ Roddy, D., *Satellite Communications*, McGraw-Hill, 4th ed., 2006.
- ⁶ Shannon, C. E., "A Mathematical Theory of Communication," *The Bell System Technical Journal*, Vol. 27, 1948, pp. 379–423.
- ⁷ Patterson, M. A., R. A. V., "GPOPS-II: A MATLAB Software for Solving Multi-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Method and Sparse Nonlinear Programming," *ACM Transactions on Math. Software*, Vol. 39, No. 3, Jul 13.
- ⁸ Patterson, M. A., R. A. V., *GPOPS-II Version 1.0: A General-Purpose MATLAB Toolbox for Solving Optimal Control Problems Using the Radau Pseudospectral Method*, University of Florida, Gainesville, FL 32611-6250, USA, 1st ed., January 2013.
- ⁹ Patterson, M. A., H. W. W. and Rao, A. V., "A ph Collocation Scheme for Optimal Control," *Optimal Control Applications and Methods*, September 2013, Submitted for Publication.

Appendix B. Conference Paper #2 (January 2015)

This appendix contains the second conference paper published as part of this research effort. It was published and presented at the AIAA 2015 SciTech Guidance, Navigation, and Control Conference in Orlando, Florida, in January 2015 [39]. This paper supersedes the first conference paper published in January 2014 A.

Optimal Airborne Trajectories for Data Collection from Wireless Sensor Networks by Direct Collocation Methods*

Nidal M. Jodeh[†] and Richard G. Cobb[‡]

Air Force Institute of Technology, Wright-Patterson AFB, OH, 45433

Riley A. Livermore[§]

Air Force Research Laboratory, Wright-Patterson AFB, OH, 45433

The optimal flight trajectory of an unmanned aerial system, collecting data from a wireless sensor network is considered. The unmanned system is tasked with collecting a minimum amount of data from each unattended and low power ground sensor in the widespread wireless sensor network. The continuous time optimal control problem solves for the optimal trajectory that minimizes the time and control effort needed to collect the required data. In this modified traveling salesman problem with neighborhoods, the control policy enables the airborne vehicle to change altitude, heading and velocity to best communicate with each ground sensor via an on-board omni-directional antenna. The optimal trajectory is solved for by a direct collocation methods also known as a pseudospectral method. The optimal path is simulated in a software-in-the-loop environment and results in a 14% decrease in flight time when compared to a traditional path planning technique.

Nomenclature

| | | | |
|----------------|--|------------------------|---|
| x | Position along the x -axis (m) | DP | Minimum Data Package Size ($bits$) |
| y | Position along the y -axis (m) | s | Number of sensors |
| z | Position along the z -axis (m) | J | Cost functional |
| \dot{x} | Velocity in x direction (m/s) | t | Time (s) |
| \dot{y} | Velocity in y directions (m/s) | α | Time scale coefficient |
| \dot{z} | Velocity in z direction (m/s) | β | Control scale coefficient |
| v | Airspeed of the UAV (m/s) | \mathbf{x} | State vector |
| θ | Pitch angle with respect to inertial frame (rad) | $\hat{\mathbf{x}}$ | UAS position vector |
| ψ | Heading angle with respect to inertial frame (rad) | \mathbf{u} | Control vector |
| $\ddot{\psi}$ | Acceleration of the UAV (m/s^2) | \mathbf{U} | Set of all admissible optimal control vectors |
| $\dot{\theta}$ | Pitch angle time rate of change (rad/s) | s | Sigmoid stiffness |
| $\dot{\psi}$ | Heading angle time rate of change (rad/s) | $\tilde{\mathbf{x}}_k$ | Position of no fly zone k |
| C_j | Bit rate for each sensor j (bps) | r_k | Radius of no fly zone k (m) |
| C_{max} | Maximum allowable bit rate (bps) | n | Number of no fly zones |
| B | Channel bandwidth (Hz) | | |
| N | Channel noise floor (W) | | |
| P | Radiated power of each sensor (W) | | |
| D_{max} | Max allowable comm distance bit rate (m) | | |
| d_j | Distance between UAV and sensor j (m) | | |
| f | Wavelength of the carrier frequency (m) | | |
| c | Speed of light (m/s) | | |

| | |
|-----|--|
| | <i>Subscripts</i> |
| j | Sensor number ($j=1,2,3,...,s$) |
| k | No fly zone number ($k=1,2,3,...,n$) |
| o | Initial |
| f | Final |

I. Introduction

The problem of determining the optimal flight trajectory for unmanned aerial system (UAS) collecting data from widespread, static and unattended, ground-based wireless sensor networks (WSN) is considered. As WSN grow in size

*Distribution C: Cleared for public release, case number 88ABW-2013-5148.

[†]PhD Candidate, Department of Aeronautics and Astronautics, Senior Member AIAA.

[‡]Associate Professor, Department of Aeronautics and Astronautics, Associate Fellow AIAA.

[§]Aeronautical Engineer, AFRL/RQQA, Member AIAA.

and their applications become more diverse, collecting their critical data becomes more challenging. UAS and WSN are currently employed in a variety of applications such as monitoring seabird habitats¹, coral reefs and fisheries², structural health³, forest fires⁴, fertilizer and pesticide applications in agriculture⁵, vehicular ground traffic in urban areas⁶, boarder patrols⁷, and tracking illegal boarder crossings[¶]. Their detailed data can provide insight into patterns never seen before, leading to more accurate models, predictions, and observations. Although the uses for WSN are limitless, the data contained in them may quickly become voluminous and difficult to collect. Efficient collection methods are needed to make these networks more useful.

Wireless sensors are ideally suited to collect localized environmental data and can be placed in austere or rugged locations. Ground-based data collection agents cannot easily reach sensor nodes in such locations. Multi-hopping data from sensors to sink requires line of sight between each node, and energy consumption rates of the inner most nodes is high.^{8,9} However the unmanned aircraft is ideally suited to collect data from sensors in hard to reach or distant locations.

II. Motivation

The scenarios investigated herein are based on research by Kingston¹⁰, Marshall et al.¹¹, and Klesh et al.¹² and are of interest to the Air Force Research Laboratory (AFRL). AFRL proposes a concept where a UAS visits all the sensors to collect required data, obviating the need for any ground-based sink. This architecture enables one UAS to serve a large number of geographically distributed sensors, freeing them from line of sight communication requirements. Specifically, AFRL's interest is in understanding localized vehicular and pedestrian traffic within a given region. Unattended and static, ground based sensors can monitor such traffic. Simply stated: given a collection of ground sensors in a WSN, how can their data be efficiently collected by an overflying vehicle?

This work updates and continues of the research by Jodeh et. al. in 2014¹³. Improvements to the optimization algorithm and plots are included. These improvements eliminated the heading correction loops experienced in the 2014 paper. Also included is a new communications model based on empirical data. Finally, optimal trajectories for a proposed flight test are presented.

Lastly, we believe this is the first time the direct collocation method is applied to determine the optimal flight path of a UAS collecting data from a WSN. This approach is novel because the optimal path is predicated on the amount of data required to be collected and is included in the optimal control problem as integral path constraints which is solved with direct collocation.

III. Problem Statement

The problem's objective is to minimize the required UAS flight time and control effort required to collect the minimum required data from a WSN. This problem can be viewed as a modified traveling salesman problem with neighborhoods,¹⁰ where the neighborhoods are each sensor's communication range. The UAS antenna is required to be within range of each sensors long enough to ensure transmission of the minimum required data.

Model and Assumptions

A. Sensor Modeling

The WSN topology is assumed homogeneous. In other words the transmission power, storage capacity, communication capability, and sensing capability of each sensor is uniform across the WSN. Static, ground based sensor locations are known to the UAS a priori and do not move once placed. Sensors are low powered, communication range limited, and often expendable. Although considered part of the network, they cannot communicate with each other due to the distance between each one, hence the use of the term widespread. They are also incapable of communicating via satellite due to power and bandwidth limitations. The overflying UAS is assumed the only means of communicating sensor data and acts as the mobile networking agent. Finally, the sensors' onboard storage capacity is considered sufficient for the duration of the scenarios.

B. UAS Modeling

The UAS is modeled as a point mass system. Control inputs are the time rate of change of heading, pitch, and velocity. The UAS size, weight, and performance specifications are limited to a Tier N/A or Tier 1 type vehicle, as defined by

[¶]Case study cited by Applied Research Associates on Jan 30, 2014 and can be found at <http://forcepro.ara.com/case-study-california-rancher-deploys-e-ugs-seismic-sensors-intercept-illegal-immigrants>, accessed on Apr 14, 2014

the US Air Force¹⁴. It is also assumed the UAS has unlimited on-board data storage and range for the duration of the test scenarios. Stochastic disturbance inputs such as wind and other random effects are not considered. Only one orbit of the WSN is calculated. Additional orbits are presumed to be multiples of the single orbit and the number of additional orbits is subject to user requirements, UAS limitations, etc. An orbit is defined as one continuous flight path circuit by the UAS within the bounds of the WSN.

C. Communications Modeling

The UAS and each sensor are equipped with a single omni-directional antenna that is attenuated only by free-space spreading loss. The sensors' transmission is continuous and isotropic over the entire field consistent with a fixed transmitted power. Two-way communication and error checking are not considered. The UAS antenna is required to be in range with each sensor long enough to ensure transmission of the specified minimum required data. The UAS is capable of simultaneously communicating with multiple sensors when in range.

IV. The Optimal Trajectory Problem

In this section, the optimal control problem, equations of motion, and communications model are developed. The dynamic model is described by the equations of motion subject to certain constraints. The pitch, heading, and velocity rates are three independent controls. The goal is to minimize the time and control effort required to collect a specified minimum amount of data from each of the sensor nodes in the WSN. The UAS is expected to be able to fly an optimal trajectory to meet the stated goals.

A. The Optimal Control Problem Definition

A continuous time optimal control problem is set up to determine the control inputs that meet the performance limits of the UAS and sensors, remain within the airspace boundaries, and minimize the flight time and control inputs. Specifically, we find the optimal control, \mathbf{u}^* , in the set of admissible controls, U , that causes the system dynamics, $\dot{x}(t) = f(x(t), u(t), t)$ to follow an optimal trajectory, \mathbf{x}^* , in the set of admissible trajectories, X , that minimizes the cost function, J , and satisfies the boundary and path constraints¹⁵. Conventionally stated by¹⁶⁻¹⁸, and without loss of generality, determine the control $\mathbf{u} \in U$, state $\mathbf{x} \in X$, initial and final time, on the interval $t \in [t_o, t_f]$ that minimize the cost functional,

$$J = \Phi(x(t_o), t_o, x(t_f), t_f) + \int_{t_o}^{t_f} g(x(t), u(t), t) dt \quad (1)$$

subject to the dynamic constraints,

$$\dot{x} = f(x(t), u(t), t) \quad (2)$$

the boundary conditions,

$$\phi(x(t_o), t_o, x(t_f), t_f) = 0 \quad (3)$$

and the inequality path constraints

$$\Omega(x(t), u(t), t) \leq 0 \quad (4)$$

Equations 1 - 4 are collectively referred to as the *continuous Bolza problem*¹⁶⁻¹⁸. In Equation 1, Φ represents the cost associated with the end points, or the "terminal cost," and g represents the cost associated with the states and controls along the trajectory, or the "running costs." In this research, the terminal cost is the final flight time and the running costs are control usage for the duration of the flight.

B. Equations of Motion

The problem is defined by six states, \mathbf{x} . The three position states of the UAS are defined along the $x(t)$, $y(t)$, and $z(t)$ axis with respect to the origin in the inertial frame. The next three states are the pitch angle, $\theta(t)$, measured positive up from the inertial xy -plane, the heading angle, $\psi(t)$, measured positive counterclockwise in the xy -plane from the inertial x -axis, and the airspeed of the UAS, $v(t)$. The controls are the pitch angle rate in the inertial frame, $\dot{\theta}(t)$, and the heading angle rate in the inertial frame, $\dot{\psi}(t)$, and acceleration of the UAS, $\dot{v}(t)$. All states and controls are time-dependent and the variable t will be dropped for the balance of the paper. The states and controls are bounded by the appropriate limits of the UAS.

The equations of motion are defined by the relationship between the states and the controls. They are derived from the geometry and differential kinematics and are given in Equation 10. A geometric definition of the variables is supplied in Figure 1 where the UAS, represented by a point mass, resides at the origin of the coordinate system. A more complex model considering roll and shadowing of the receiver's antenna may be considered for future research.

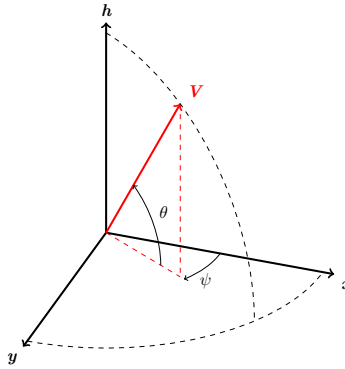


Figure 1. State variables in relation to the inertial frame coordinate axes.

C. Communications Model

While our emphasis in developing an optimal trajectory, it is necessary to include discussion on the communications model. To formulate the optimal path, the optimizer requires only a data rate for a given euclidean distance, d , between the ground sensor and the UAS. The data rate varies as a function of the distance, and optimizing the data rate to meet constraints is the primary dynamic relationship used to determine the optimal trajectory for the UAS. The distance from the UAS to sensor j is expressed in terms of the state variables and the j^{th} sensor's position coordinates, Equation 5.

$$d_j = \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad \forall j \quad (5)$$

Because the communication is intended to collect information stored in the sensor's memory, the parameter of interest is the total volume of data transferred, measured as the number of bits. The volume of data received over the time interval is then the integral sum of the data rate. The user sets the minimum data required, thus establishing an integral constraint in the optimal control problem formulation.

A traditional client and server architecture is employed, where the UAS acts as the server and each of the ground sensors are clients who communicate with the UAS when in range. This "hub and spoke" model allows multiple ground sensors to connect with the UAS and exchange information simultaneously. The UAS will connect when within the neighborhood of any sensor. Two communications models were considered in this research, the Shannon-Hartley model and an empirically based emulation.

1. Shannon-Hartley Model

The physics-based, Shannon-Hartley theorem describes the theoretical communication channel capacity. In other words, the channel capacity is the maximum data rate that information can be sent over a certain frequency, f , in the presence of noise, N , with a given bandwidth, B , and transmitter power, P . Equation 6 determines the maximum theoretical data rate, or the channel capacity, C , for any given sensor, j , as a function of the distance, d , from the UAS to that particular sensor.

$$C_j = B \log_2 \left(1 + \frac{P}{N \left(\frac{4\pi d_j f}{c} \right)^2} \right) \quad (6)$$

The Shannon-Hartley theorem was not chosen for this research because of the uncertainty in the radio parameters and the unnecessary complexity of the model. To effectively implement Shannon-Hartley, the power settings, frequency, bandwidth, and noise must all be known during the UAS flight time. While the communication frequency is set at 2.4GHz, the power and bandwidth are variables the radio software manipulates to maintain a strong signal. Additionally, there are unknown variations in the environmental noise. Determining representative noise, bandwidth and

power settings for the communication environment is not a trivial task. A complex communication model is unrealistic for this research and all of the uncertainty involved in identifying the communication environment parameters renders the Shannon-Hartley theorem infeasible.

2. Empirical Communications Model

A second model was developed which is considerably simpler and based on empirical data from AFRL flight tests. This empirical approach merely emulates the observed behavior of the communications between the AFRL sensors and UAS without modeling the radio hardware. It was observed that the information data rate remains constant over the entire connectivity range. Now, instead of four variables required to determine the communications model, only the maximum connectivity distance is required to have an accurate and representative real-world communications model.

The ground sensors are programmed to transmit messages with a fixed data package size at a constant frequency. When the UAS is within the ground sensor's communication range, it receives data at a constant rate. By maintaining a constant data rate, the only variable becomes whether the sensor and UAS are connected or disconnected. Determining the range where the connection breaks is a tractable problem that can be done with ground and flight test. The essence of the empirical communication model is determining the on/off connection range for a set data rate, independent of the communication environment.

In 2014, AFRL conducted three flight tests of their UAS and sensor hardware. The data size and transmit frequency were set to a constant data rate of 5 kbit/s. This data rate is far below the available bandwidth to ensure that multiple ground sensors can connect to the UAS simultaneously. A series of varying range tests were conducted to determine the maximum line of sight communication range. The UAS consistently maintained its connection with the ground sensor within a distance of at least 1 km. On several occasions, the UAS continued to receive the constant 5 kbit/s rate for up to two kilometers. However this was not typical. The maximum connection range was conservatively set to 1 km to ensure that the UAS would always be connected with the ground sensor within that range. AFRL's communication setup features the WiFi 802.11b protocol and behaves in a traditional client and server manner.

As expected, the observed communication behavior toggled between two discrete options, connected or disconnected. When connected, a constant rate of 5 kbit/s was observed. For implementation into the optimal control problem, this discrete behavior needed to be transformed into a continuous and differentiable function. A sigmoid function, represented in Equation 7 was used because it satisfied the continuous and differentiable requirements, but also allowed for a variation of the stiffness without losing gradient information away from the cutoff value, critical for optimizer search directions. The stiffness feature allowed for approximation of a step function, or the behavior observed. As the stiffness, s , is increased, the sigmoid function approaches a step function as shown in Figure 2(a)

$$C_j = \frac{C_{max}}{1 + e^{-s(D_{max}-d_j)}} \quad (7)$$

Where d_j is from Equation 5, and C_{max} and D_{max} are the maximum communications rate of 5 kbit/s and distance 1 km respectively. Figure 2 illustrates the empirical model with observed data rates. Two stiffness values, s , were used to illustrate how the model can be modified to mimic the discrete behavior of data rate past a given distance, in this case 1 km.

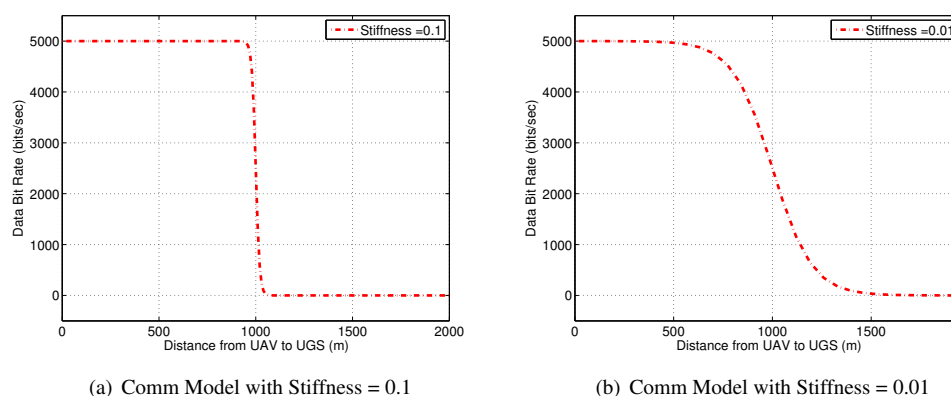


Figure 2. The empirical communications model with a maximum data rate of 5 kbit/s and range of 1km for two stiffness values, $s=0.01$ and $s=0.001$

Unfortunately, increasing the stiffness also steepens the gradients in the region surrounding the maximum communication range D_{max} . These steep gradients make the optimizer more reliant on the initial guess, making search making search directions difficult for the optimizer. Without sufficient gradient information, solutions are less likely to converge. Selecting a sigmoid stiffness of 0.01 provided enough gradient information for the optimizer to converge to the optimal solution while closely mimicking the empirical behavior.

The optimal path formulation is agnostic to the particular communication model, so long as the data rate is a function of distance. Although both communication models are able capable of this task, this research uses the empirical model for its ease and representation of the observed communication.

D. No-Fly Zones

To model airspace or terrain limitations, no-fly zones were also incorporated. The no-fly zones are represented as path inequality constraints in the optimal control problems. Simply stated, the difference between the position states of the UAS, denoted as $\hat{\mathbf{x}}(t)$ to distinguish if from the full state vector, and the center of the no-fly zone, $\tilde{\mathbf{x}}_k$, must be greater than or equal to radius, r_k , of the k^{th} no-fly zone.

$$\|\hat{\mathbf{x}}(t) - \tilde{\mathbf{x}}_k\|_2 \geq r_k \quad \forall k \quad (8)$$

E. Optimal Control Problem Formulation

With the system dynamics and the integral path constraint, the formal definition of the continuous optimization problem is complete. The solution is the control which minimizes the cost function, Equation 9, subject to the system dynamic, Equation 10, the boundary conditions, Equation 12, and the path and integral constraints in Equations 13 and 14. Where the path and integral constraints represent the specific formulation of Equation 4. Time and control weighting factors, α , and β , respectively, are use to tune the influence of the individual components of the cost function.

The optimal control problem is now formulated to find the optimal control, \mathbf{u}^* , in the set of admissible controls, U , that minimizes the following cost functional

$$\min_{\mathbf{u} \in U} J = \alpha(t_f) + \int_{t_0}^{t_f} \beta_1 \dot{v}^2 + \beta_2 \dot{\theta}^2 + \beta_3 \dot{\phi}^2 dt \quad (9)$$

subject to the dynamic constraints

$$\dot{x} = v \cos \theta \cos \psi \quad (10a)$$

$$\dot{y} = v \cos \theta \sin \psi \quad (10b)$$

$$\dot{z} = v \sin \theta \quad (10c)$$

where the states and controls are

$$\mathbf{x} = [x, y, z, \theta, \psi, v]^T \quad (11a)$$

$$\mathbf{u} = [\dot{\theta}, \dot{\psi}, \dot{v}]^T \quad (11b)$$

with the boundary conditions,

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (12a)$$

$$\mathbf{x}(t_f) = \mathbf{x}_f \quad (12b)$$

the path constraints ,

$$\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \quad (13a)$$

$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max} \quad (13b)$$

$$\|\hat{\mathbf{x}}(t) - \tilde{\mathbf{x}}_k\|_2 \geq r_k \quad \forall k \quad (13c)$$

and the integral constraints

$$\int_{t_0}^{t_f} C_j dt \geq DP_j \quad \forall j. \quad (14)$$

V. Solution

A. Problem Specific Values

UAS performance limits were derived from various sources and consistent with nominal USAF Tier 1 UAS and commercial small UAS in use today.^{14||} In other cases, AFRL specific flight test parameters are used when verifying flight test scenarios. For example, AFRL flies UAS as a fixed airspeed of 14 m/s. Specifications selected for the sensor model are also based on empirical data. Table 1 lists limits and constants imposed throughout the test cases.

Table 1. Problem Values

| Parameter | Values |
|-----------------------|---|
| Pitch | $-18 \leq \theta \leq 18 \text{ (deg)}$ |
| Heading | $-180 \leq \psi \leq 180 \text{ (deg)}$ |
| Velocity | $10 \leq v \leq 30 \text{ (m/s)}$ |
| Pitch Rate | $-25 \leq \dot{\theta} \leq 25 \text{ (deg/sec)}$ |
| Heading Rate | $-6 \leq \dot{\psi} \leq 6 \text{ (deg/sec)}$ |
| Acceleration | $-1 \leq \dot{v} \leq 1 \text{ (m/s}^2\text{)}$ |
| Maximum Data Rate | $C_{max} = 5 \text{ kbits/s}$ |
| Maximum Comm Distance | $D_{max} = 1 \text{ km}$ |
| Stiffness | 0.01 |

B. Initial Guess

An initial guess was provided to the optimizer for each optimal solution, also referred to as known as a “warm start.” Initial guesses are required if stiffness values are high enough such that gradient information is negligible to the optimizer across the entire WSN. Stiffness values ≥ 0.01 require an initial guess. Furthermore, initial guesses are made without no-fly zones or other constraints that may be imposed on the desired optimal path. The initial guess helps to satisfy the assumption that the optimizer knows the locations of the sensors a priori. The initial guess was determined with the exact same optimal control formulation, only using a relaxed stiffness value, enabling the optimizer to detect sufficient gradient information to locate each sensor. Stiffness values ≤ 0.001 were used to determine the first solution, known as a “cold start.” This first value was saved and fed back to the optimizer, but this time with the higher stiffness value. Initial guesses are shown in each of the test case plots.

C. Cost Functional Weightings

The cost functional weightings α and β are used to scale and tune the optimal trajectory. This was largely done heuristically, using engineering judgment after examining the initial solutions. An initial setting for all the test cases began with $\alpha = 0.01$ and $\beta = [1, 1, 0.001]$. These initial settings were chosen to scale the final time, smooth out the flight path, and encourage velocity changes. However these values are not intuitive and can be updated with successive solutions. Parameter optimization should be considered for future work.

D. Optimal Trajectory Test Cases

To test and demonstrate the solution method’s capability, optimal trajectories were determined for various scenarios. Test cases are tabulated in Table 2. Variations in test cases included sensor locations, elevations, minimum data requirements, take-off/landing or orbiting, and initial guesses. Each of the following examples increase in complexity and dimension. Two-dimensional, single sensors cases are used primarily as baseline checks of the optimization routine, where the solution is easily interpreted. In these scenarios, the UAS and sensors are fixed at the same altitude.

In the first example, the UAS is required to depart and return to an airfield, taking off and landing on the same heading. For the single sensor test case, two differing data requirements were imposed. In the second example, the UAS is already airborne only required to fly the optimal path between four sensors. One of the four sensors is elevated

^{||}Reference to the MLB Corp SuperBAT and BAT 4 UAS for example, <http://spyplanes.com/products-s-bat/>

to illustrate the three-dimensional and velocity control capability. These first two examples were chosen for their simplicity and expected solutions. In the next test case, a random scatter of 10 sensors is used with a no-fly zone constraint. Two differing initial guesses are used on the set of 10 sensors to test the response of the optimizer. Finally, a test case called “Atterbury,” uses actual sensor positions from a proposed future flight test. The Atterbury test case compares a point to point solution versus the optimal solution.

No upper limit was imposed on the amount of data that can be collected. Enforcing an upper limit over-constrains the problem and forces the optimizer to skirt the outer bounds of the communication range to prevent violating any upper data collection limits. In reality, if the sensor has uploaded all of its data, the UAS continues along a path towards the next sensor, and that may include overflight of a region of maximum data rates.

Table 2. Optimal Trajectory Test Cases

| Test Case I.D. | No. of Sensors | Min Data Required (<i>kbit</i>) | Take-Off/Landing or Orbit |
|-------------------|-------------------|--------------------------------------|------------------------------|
| 1a | 1 | 10 | Take-Off/Landing |
| 1a | 1 | 600 | Take-Off/Landing |
| 2 | 4 | 500 | Orbit |
| 3a | 10 | 10 | Orbit |
| 3b | 10 | 10 | Orbit |
| Atterbury | 10 | 730 | Orbit |

E. Solution Method

The MATLAB-based, General Purpose Pseudospectral Optimal Control Software-II (GPOPS-II) is the program used to solve for the optimal trajectories. GPOPS-II directly solves the continuous time optimal control problem.¹⁹ GPOPS-II transcribes the continuous-time optimal control problem into a discrete finite-dimensional static optimization problem that is solved as a Non-Linear Programming (NLP) problem. GPOPS-II is implemented using the *hp*-adaptive version of the Radau pseudospectral method. This particular method is an orthogonal collocation Gaussian quadrature implicit integration method in which the collocation is performed at the Legendre-Gauss-Radau quadrature points.²⁰ An adaptive mesh refinement method is implemented that determines the number of mesh intervals and the degree of the approximating polynomial for each mesh interval in order to satisfy the specified error tolerance.²⁰ GPOPS-II can be set to prioritize finding the degree of the approximating polynomial before refining the mesh interval or vice versa.

F. GPOPS-II Settings

For all the test cases, GPOPS-II is set to use the IPOPT NLP solver. The derivative approximation used is “sparseCD” and second derivatives are provided. The mesh method is set to “hp-LiuRao,” where GPOPS-II increases the degree of the approximating polynomial up to a maximum allowable degree prior to dividing the interval into sub-intervals or combine sub-intervals in larger intervals.²¹ The NLP solver’s error tolerance is set to 10^{-6} . “RPM-integration” is enabled and “automatic-hybridUpdate,” scaling is performed. All results presented are from computations performed using MATLAB 2012b on a 2012 MacBook Pro with a 2.9 GHz Intel Core i7 processor and 8 GB of 1600 MHz DDR3 memory.

VI. Results

Various results of the optimal control problem are presented in this section. The following plots show the optimal solution to each specified scenario in Table 2. Furthermore, all solutions met the optimality criteria and error tolerances. The results are presented to better illustrate to the reader the optimal solution in a layered method, from simple to complex. First, the two-dimensional cases for a single sensors are presented. Next, a three-dimensional case for four sensors is presented. A more complex example illustrates the solution for a UAS communicating with 10 randomly placed sensors in a WSN with a no-fly zone constraint. Concentric rings of 99%, 50%, and 1% of maximum data rates from the sensor are displayed as top down views of the communications model to indicate when UAS is in range of the particular sensor. The final test case, Atterbury, is discussed in Section VII..

A. Single Sensor Test Cases

1. Single sensor Test Cases 1a and 1b - Results

Figure 3 illustrates the optimal flight path of a UAS required to take off, collect 10 and 600 kbit of data respectively from a single sensor, and return to land at the airfield. The airfield resides at the origin, represented by waypoint 1 on the flight path plots. Also included in the flight path plots are the percent data rates as a function of distance from the sensor. This is a top down view of the sigmoid communication model in Figure 2. The initial guess, denoted by a line with diamond markers, satisfies the requirement for a priori knowledge of sensor location. The resulting data collection rates are presented in Figure 4, and the required pitch, heading, and velocity in Figure 5. These detailed plots are provided for the reader's understanding and will not always be presented. The legend in Figure 3(c) applies to Figures 3 and 6

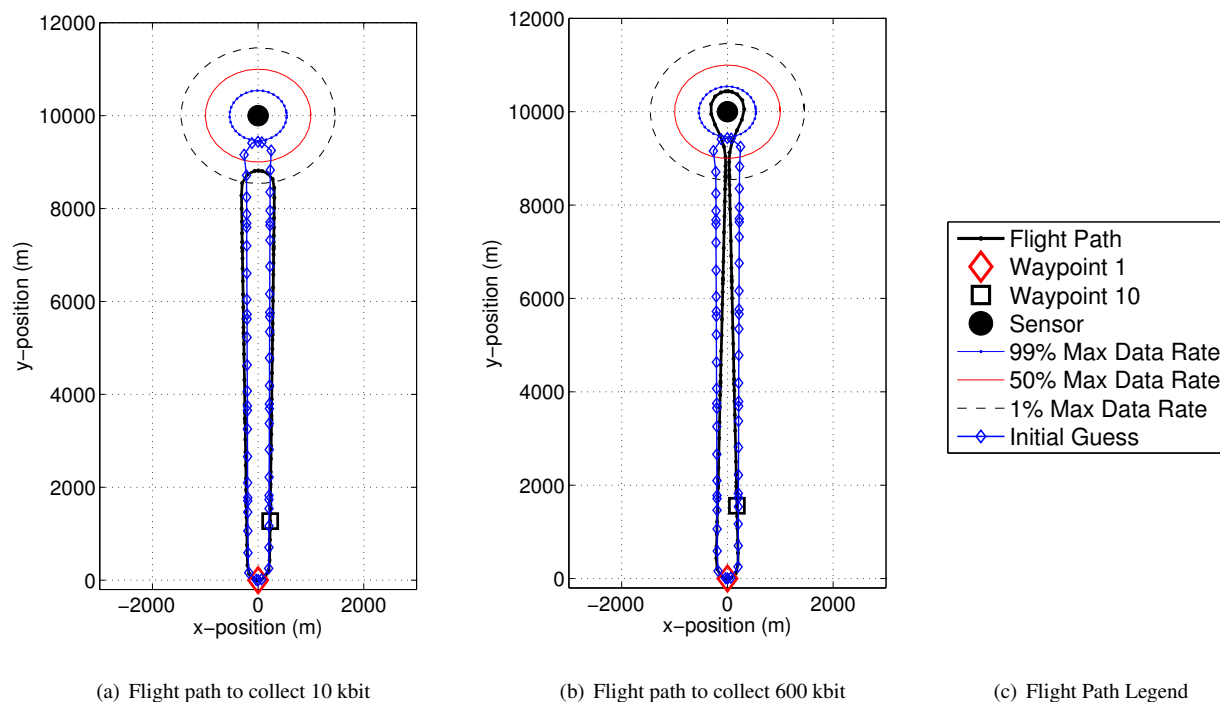


Figure 3. Optimal UAS trajectories to collect 10 and 600 kbit of data from a single sensor respectively. The data rates as a function of distance from the sensor and initial guess are also plotted

The effects of changing the data requirements clearly result in changes to the optimal flight path and data rates. These simple examples help to illustrate the changing solutions as a result of increased demand. Case 1a resulted in a flight time of 613 seconds. By comparison, case 1b required 735 seconds to complete, thus resulting in a longer flight path around the sensor in Figure 3(b).

The data rates are also reflected in Figure 4 and confirm the integral constraint. As expected the data rate increases as the UAS approaches the sensor, and decreases as it flies further away. In Figures 4(a) and 4(b), total area under the plot is equal to the minimum data required of 10 and 600 kbit respectively.

The UAS pitch, heading, and velocity states shown in Figure 5 also confirm the flight path plots. After takeoff, UAS accelerates to the maximum velocity for the duration of the scenario. However in the 600 kbit solution in Figure 5(b), the optimizer required a slower velocity to ensure it collected the required data. The velocity change is clearly seen at the same time of the most significant heading change, indicating it slowed down at the sensor to take advantage of the maximum data rates.

B. Four Sensor Test Cases

Figure 6 illustrates four sensor nodes are placed at the corners of a square area, with one sensor elevated 2,000 meters above the surface. As in the single sensor case, this arrangement is chosen because the solution is assumed to be

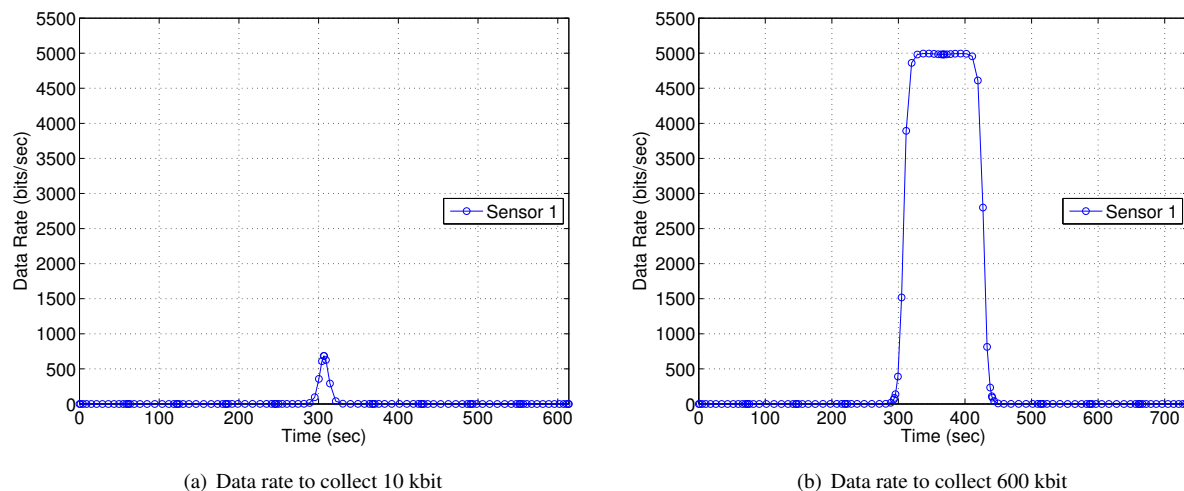


Figure 4. Data rates from each of the respective data collections in Figure 3

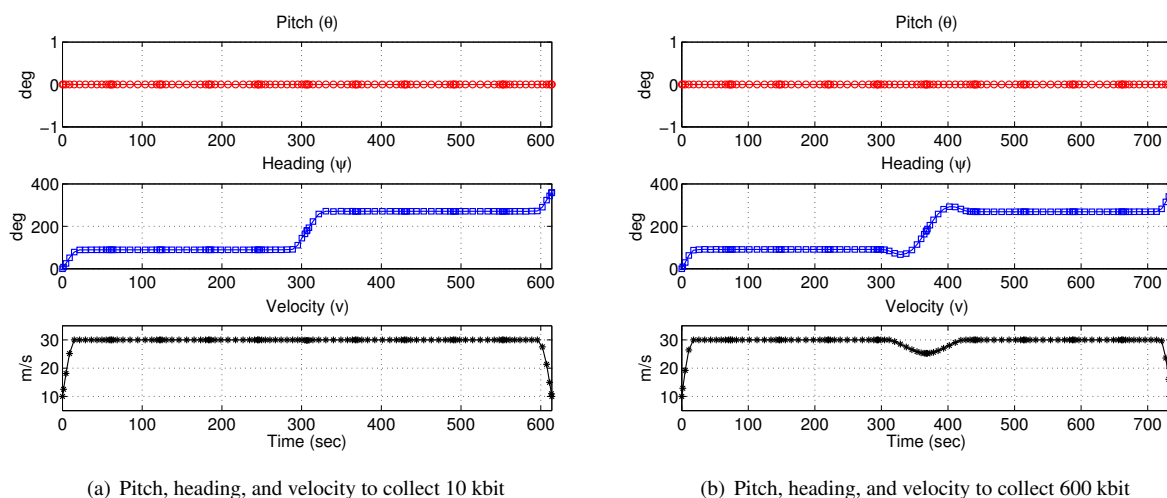


Figure 5. Pitch, heading and velocity from each of the respective data collections in Figure 3.

intuitive. In this test cases the UAS is required to collect 500 kbit of data from each sensor. This example was chosen to examine the three-dimensional capability of the optimizer.

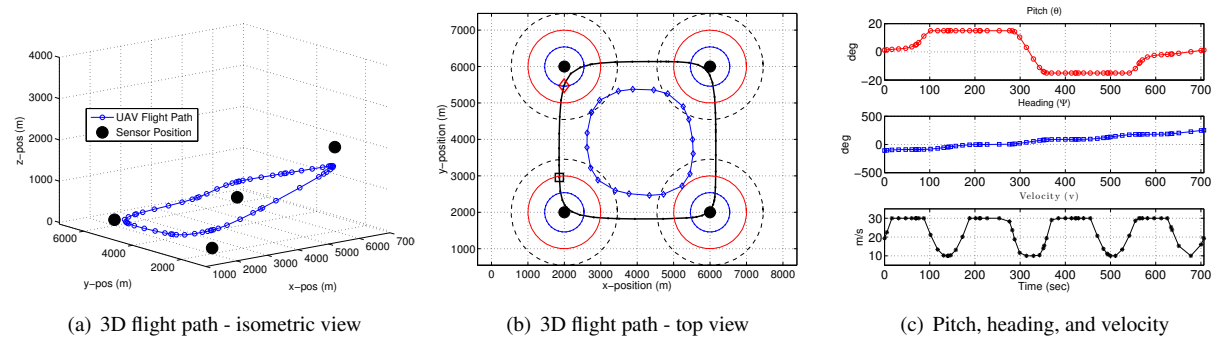


Figure 6. Three-dimensional, optimal UAS trajectory for a four sensor test case where one sensor is elevated 2,000 meters. Refer to the legend in Figure 3(c)

In the four sensor test case, the UAS collected the minimum required 500 kbit of data. Since no minimum altitude was set, data was allowed to be collected from below the sensor as seen in Figure 6(a). Control usage was more pronounced in this scenario as pitch and velocity varied significantly in Figure 6(a). The optimizer employed a slower velocity over the sensors to ensure the minimum data was collected, especially in the region of high data rates (blue concentric ring) at each sensor.

C. Ten Sensor Test Case

Figure 7 illustrates the 10 random sensor test case. In this scenario, some sensors are placed around the no-fly zone that represents a mountain or other obstruction. The UAS was tasked with collecting 100 kbit of data from each sensor. Two different initial guesses were used to determine their impact on the optimal path.

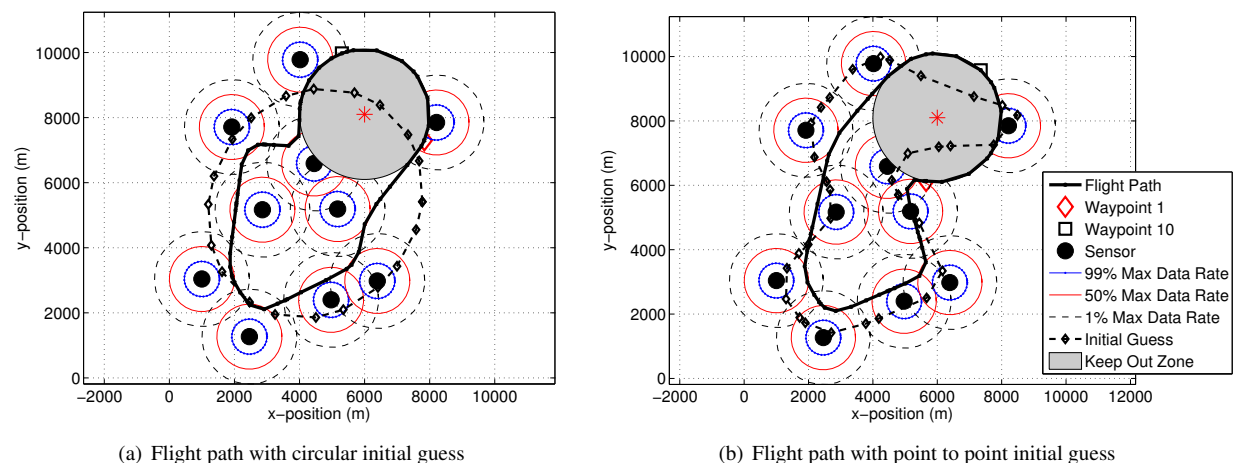


Figure 7. Optimal UAS trajectory for 10 random sensors with keep out zone and initial guess

Despite two different initial guesses, optimal solutions were found for both inputs. Figure 7(a) shows the initial guess as a more circular path, which resulted in an optimal flight time of 751 seconds. Figure 7(b) shows the initial guess as a point to point path. This resulted in an optimal flight time of 776 seconds, 25 seconds longer than the optimal path resulting from the circular initial guess, or a 3% difference.

In addition to comparing the final time of the two optimal flight paths, the total amount of data collected can also be examined. Figure 8 displays the data collected as a result of the different initial guesses. The circular initial guess resulted in an optimal path that collected 100 kbit from six sensors. From the remaining four sensors, 340, 231, 180, and 251 kbit were collected respectively. The total data collected as a result of the circular initial guess was 1604 kbit.

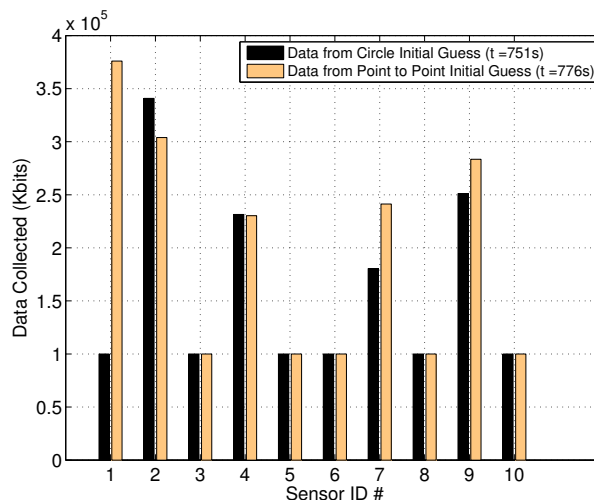


Figure 8. Data collected from the ten sensor scenario for the circular and point to point initial guesses

Using the point to point initial guess resulted in an optimal path that collected a 100 kbit from five sensors. The UAS also collected 376, 303, 230, 241, and 283 kbit from the remaining five sensors respectively. The total data collected using the point to point initial guess was 1935 kbit, or 331 kbit more than the optimal path that resulted from the circular initial guess.

Clearly the initial guess had an influence on the optimal path. The two results represent local optimal solutions and may not be the global optimal solution. Factors such as cost functional weights, total data required, maximum communication range and rates, stiffness, and UAS performance parameters are all subject to change. These values depend largely on the hardware limitations, mission parameters, and user desires. Tighter tolerances on the total data collected for each could be imposed, however based on our experience, this tends to over-constrain the problem and sometimes prevent it from converging. The cost functional weights also contribute to the optimal flight path. Although they were fixed for both initial inputs in this ten sensor scenario, future work should consider a parameter optimization for the cost functional weights and seek to minimize the impact of the initial guess.

VII. Simulation

High fidelity simulations are the next logical step to verify the optimal paths calculated using GPOPS-II. Ultimately, the final goal of this research is to validate the MATLAB generated optimal paths using real-world flight test. The AVTAS** Multi-Agent Simulation Environment (AMASE) suite is used to simulate the UAS flying the calculated optimal path.²² This section briefly details the inner workings of AMASE and presents the simulation results from flying the GPOPS-II calculated optimal paths. Understanding how AMASE works requires knowledge of the foundational software architecture. The Common Mission Automation Services Interface (CMASI) defines the underlying protocol that allows all systems to interface seamlessly.²² AMASE uses CMASI extensively and is an all-in-one simulation environment that can perform both software and hardware in the loop testing. AFRL's Controls Science Branch uses the AMASE platform to verify UAS control algorithms, mission objectives, and system interoperability, prior to actual flight test^{††}. A strong correlation exists between simulation and real-world flight test success due to the robustness of AMASE.

For this research, AMASE is used as an intermediate step prior to any flight testing. Employing AMASE in this manner has several benefits. The first ensures GPOPS-II results are properly translated into CMASI messages with pertinent information for the UAS to fly the desired path. Without properly translating the data into the correct XML format messages, the optimal path cannot be flown in AMASE or transmitted to an airborne UAS. If the path can be flown in AMASE then the data are correctly formatted and capable of being transmitted and flown in real-world flight test. The second benefit of using AMASE is that the data collected from the simulation can be analyzed and evaluated independent of MATLAB. This separate evaluation verifies the MATLAB results, ensuring the reality of the optimal paths generated. Lastly, AMASE allows for a large quantity and variety of cases to be simulated in a representative

** Air Vehicle Test Analysis and Simulation

†† Based on discussion with AFRL Controls Science Branch engineers

environment. Using AMASE gives credibility to the GPOPS-II results as well as viability for real world flight test.

1. Simulation Results

In preparation for planned flight tests in 2015, the AFRL Controls Sciences Branch proposed ten sensor locations at the Camp Atterbury Training Center, Indiana. Due to access and airspace limitations, the sensors are spatially closer together than desired, but will nevertheless provide valuable insight into the data collection scheme. For the proposed flight test, a point to point solution was flown in AMASE and a minimum of 730 kbits were collected from each sensor. The point to point solution was representative of the current method of flight path planning. The same sensor locations were entered into GPOPS-II and the minimum amount of data required was set to 730 kbits for each sensor. An optimal flight path was calculated and entered back into AMASE for comparison. A representative constant altitude of 30.5 m and velocity of 14 m/s were programmed into the optimal trajectory solution and AMASE. As in the earlier test cases, the maximum data rate, distance, and stiffness were set to 5 kbits per second, 1 km, and 0.01 respectively. Figure 9(a) compares the optimal path to the point to point path and Figure 9(b) shows the amount of data collected at each of the 10 ground sensors for the GPOPS-II, AMASE and point to point paths.

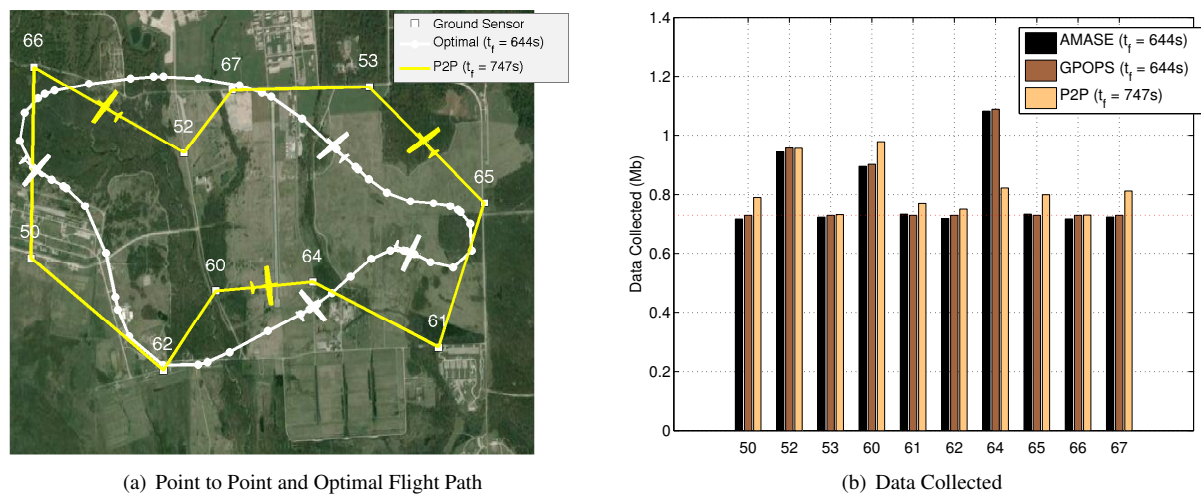


Figure 9. Comparison of a Point to Point and Optimal Flight Path collecting 730 kbit of data

2. Simulation Discussion

The collocation points in the optimal solution, produced by GPOPS-II, contain the UAS position, heading, altitude, and airspeed information, making them ideal waypoints for the UAS. In AMASE, the UAS flies the optimal path via the collocation points (line and circle path in Figure 9(a)) as waypoints while the point to point path uses each ground sensor as a waypoint. The optimal path is nearly 100 seconds faster, constituting a 14% savings in flight time while meeting the minimum data collection requirements.

Simply observing the flight paths does not give an intuitive understanding of the amount of data collected. Figure 9(b) compares the total amount of data collected via the MATLAB calculated optimal path, AMASE optimal path, and the point to point paths. All three models use the empirical communication model approximated by the sigmoid function. There are two important takeaways from Figure 9(b). The first is the data collected from the optimal path in both GPOPS-II and AMASE are nearly identical. This results makes sense because the waypoints and the communication model used for both the Matlab and AMASE environments are the same. Achieving near identical data collection is important because it verifies that the optimal path solved using GPOPS-II can be accurately translated into the XML format and simulated in AMASE. The small differences between the two are attributed to the different integration methods (Gaussian quadrature vs Runge-Kutta methods) used for calculating the total data collected. The second takeaway is the overall data collected from the optimal path achieves the minimum requirement for each ground sensor and has comparable data collection to the point to point. The advantage of flying the optimal path is that the time to accomplish the mission is minimized without a degradation of performance.

VIII. Conclusion

The optimal flight trajectory for a UAS collecting data from a widespread, static and unattended, ground-based WSN was demonstrated. An optimal control problem was set up to select the best pitch, heading, and velocity to collect the required data. The trajectories are all plausible solutions any UAS is capable of flying. However, choosing other settings, like the cost function weightings, α and β proved more difficult. Future work should consider a parameter optimization method to select these terms. As the cost function weightings are determined, sensitivity to initial guesses should be done as well. Although optimal solutions were found, their differing paths and data collected suggested local, not global, minimums. Proper translation of the GPOPS-II solution to XML in AMASE ensure the optimal path is ready for flight test. Data from flight test will aid in determining which settings are the most appropriate. A comparison of the data collected in flight against the expected amount would be a significant step to validate the optimal solution and test the XML file conversion in AMASE.

Acknowledgement

The authors would like to especially recognize and thank Dr Derek Kingston and Mr Steve Rasmussen of the Air Force Research Laboratories, Control Science Center of Excellence, who graciously provided us with the challenging problem concept, background information, and endless support.

References

- ¹Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J., "Wireless sensor networks for habitat monitoring," *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ACM, 2002, pp. 88–97.
- ²Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., and Corke, P., "Data collection, storage, and retrieval with an underwater sensor network," *Proceedings of the 3rd international conference on Embedded networked sensor systems*, ACM, 2005, pp. 154–165.
- ³Mascareñas, D., Flynn, E., Todd, M., Park, G., and Farrar, C., "Wireless sensor technologies for monitoring civil structures," *Sound and Vibration*, Vol. 42, No. 4, 2008, pp. 16–21.
- ⁴Aslan, Y. E., Korpeoglu, I., and Ulusoy, Ö., "A framework for use of wireless sensor networks in forest fire detection and monitoring," *Computers, Environment and Urban Systems*, Vol. 36, No. 6, 2012, pp. 614–625.
- ⁵Façal, B. S., Costa, F. G., Pessin, G., Ueyama, J., Vieira, H. F., Colombo, A., Fini, P. H., Villas, L., Osório, F. S., Vargas, P. A., et al., "The Use of Unmanned Aerial Vehicles and Wireless Sensor Networks for Spraying Pesticides," *Journal of Systems Architecture*, 2014.
- ⁶Xiao, Q., Xiao, B., Luo, J., and Liu, G., "Reliable navigation of mobile sensors in wireless sensor networks without localization service," *Quality of Service, 2009. IWQoS. 17th International Workshop on*, IEEE, 2009, pp. 1–9.
- ⁷Sun, Z., Wang, P., Vuran, M. C., Al-Rodhaan, M. A., Al-Dhelaan, A. M., and Akyildiz, I. F., "BorderSense: Border patrol through advanced wireless sensor networks," *Ad Hoc Networks*, Vol. 9, No. 3, 2011, pp. 468–477.
- ⁸Gu, Y., Zhao, B.-H., Ji, Y.-S., and Li, J., "Theoretical treatment of target coverage in wireless sensor networks," *Journal of Computer Science and Technology*, Vol. 26, No. 1, 2011, pp. 117–129.
- ⁹Castañó, F., Rossi, A., Sevaux, M., and Velasco, N., "On the use of multiple sinks to extend the lifetime in connected wireless sensor networks," *Electronic Notes in Discrete Mathematics*, Vol. 41, 2013, pp. 77–84.
- ¹⁰Kingston, D., "Intruder Tracking Using UAV Teams and Ground Sensor Networks," Tech. rep., Control Science Center of Excellence, Air Force Research Labs, Wright Patterson AFB, OH, USA, 2011.
- ¹¹Marshall, C., Mears, M., Kingston, D., and Rasmussen, S., "2010 ICE-T Cooperative Control Flight Testing," *Infotech@ Aerospace 2011*, No. AIAA 2011-1486.
- ¹²Klesh, A. T., Kabamba, P. T., and Girard, A. R., "Path Planning for Cooperative Time-Optimal Information Collection," *2008 American Control Conference*, 2008.
- ¹³Jodeh, N. M., Coon, T. E., Masternak, T. J., Cobb, R. G., and Agte, J. S., "Optimal Airborne Trajectories for Data Collection from Emplaced Ground Sensor Arrays," January 2014.
- ¹⁴"Unmanned Systems Integrated Roadmap FY 2011-2036," No. ADA558615, Office of the Under Secretary of Defense (Acquisition Technology and Logistics), Washington DC, Oct 2011.
- ¹⁵Kirk, D., *Optimal Control Theory, An Introduction*, Dover Publications Inc., 1998.
- ¹⁶Ross, S., *Stochastic Real-Time Optimal Control: A Pseudospectral Approach for Bearing-Only Trajectory Optimization*, Ph.D. thesis, Air Force Institute of Technology, 2011.
- ¹⁷Garg, D., Patterson, M. A., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "An Overview of Three Pseudospectral Methods for the Numerical Solution of Optimal Control Problems," 2009.
- ¹⁸Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435–1440.
- ¹⁹Patterson, M. A., R. A. V., "GPOPS-II: A MATLAB Software for Solving Multi-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Method and Sparse Nonlinear Programming," *ACM Transactions on Math. Software*, Vol. 39, No. 3, Jul 13.
- ²⁰Patterson, M. A., R. A. V., *GPOPS-II Version 1.0: A General-Purpose MATLAB Toolbox for Solving Optimal Control Problems Using the Radau Pseudospectral Method*, University of Florida, Gainesville, FL 32611-6250, USA, 1st ed., January 2013.
- ²¹Patterson, M. A., H. W. W. and Rao, A. V., "A ph Collocation Scheme for Optimal Control," *Optimal Control Applications and Methods*, September 2013, Submitted for Publication.
- ²²Duquette, M., "The Common Mission Automation Services Interface," *Infotech@ Aerospace 2011*.

Appendix C. Conference Paper #3 (January 2015)

This appendix contains the third conference paper published as part of this research effort. It was published and presented at the AIAA 2015 SciTech Intelligent Systems Conference in Orlando, Florida, in January 2015 [128].

Autonomous Flight Path Planning for Traffic Monitoring in Wireless Sensor Networks*

Nidal M. Jodeh[†] and Richard G. Cobb[‡]

Air Force Institute of Technology, Wright-Patterson AFB, OH, 45433

In this research, ground-based wireless sensor networks and unmanned aircraft are used to determine optimal flight paths to ensure longer dwell times over areas of high activity. Sensor nodes placed in rugged, austere, or dangerous locations may be difficult to access. Ground-based data collection agents can be limited by range and terrain. Multi-hopping data between sensor nodes are subject to communication range limits, while energy reserves of nodes handling the highest data traffic decay at a faster rate. The unmanned aircraft is ideally suited to overcome these limitations. The scenario proposed in this research utilizes the unmanned aircraft to fly an optimal flight path within the wireless sensor network, collect their data, determine which sensor nodes are experiencing the most activity in their sensing range. Once the sensor nodes with the highest activity rates are identified, a new optimal flight path is calculated to ensure longer flight times over those particular sensors, while maintaining or decreasing flight times over sensors with less or no activity. When multiple unmanned aircraft are employed, sensors are clustered by euclidean distances to a centroid by a k -means clustering algorithm. Optimal flight paths are determined by solving an optimal control problem using direct collocation methods. Single and multiple unmanned aircraft solutions are presented.

Nomenclature

| | | | |
|----------------|--|------------------------|---|
| x | Position along the x -axis (m) | t | Time (s) |
| y | Position along the y -axis (m) | α | Time scale coefficient |
| z | Position along the z -axis (m) | β | Control scale coefficient |
| \dot{x} | Velocity in x direction (m/s) | \mathbf{x} | State vector |
| \dot{y} | Velocity in y directions (m/s) | $\hat{\mathbf{x}}$ | UAS position vector |
| \dot{z} | Velocity in z direction (m/s) | \mathbf{u} | Control vector |
| v | Airspeed of the UAV (m/s) | \mathbf{U} | Set of all admissible optimal control vectors |
| θ | Pitch angle with respect to inertial frame (rad) | s | Sigmoid stiffness |
| ψ | Heading angle with respect to inertial frame (rad) | $\tilde{\mathbf{x}}_k$ | Position of no fly zone k |
| \ddot{v} | Acceleration of the UAV (m/s^2) | r_n | Radius of no fly zone n (m) |
| $\dot{\theta}$ | Pitch angle time rate of change (rad/s) | K | Number of UAS |
| $\dot{\psi}$ | Heading angle time rate of change (rad/s) | N | Number of Fly/No Fly zones |
| C_j | Bit rate for each sensor j (bps) | <i>Subscripts</i> | |
| C_{max} | Maximum allowable bit rate (bps) | j | Sensor number ($j=1,2,3,...,s$) |
| D_{max} | Max allowable comm distance bit rate (m) | n | Fly/No Fly zone number ($n=1,2,3,...,N$) |
| d_j | Distance between UAV and sensor j (m) | k | UAS number ($k=1,2,3,...,K$) |
| DP | Minimum Data Required ($bits$) | o | Initial |
| s | Number of sensors | f | Final |
| J | Cost functional | | |

I. Introduction

In the foreseeable future, unmanned aircraft systems (UAS) can reduce manpower costs required to accomplish various tasks. Combined with ground-based wireless sensor networks (WSN), the UAS's ability to monitor activity on the ground is significantly enhanced. However, algorithms are needed to enable UAS to optimally and autonomously

*Distribution C: Cleared for public release, case number 88ABW-2014-5761.

[†]PhD Candidate, Department of Aeronautics and Astronautics, 2950 Hobson Way, Senior Member AIAA.

[‡]Associate Professor, Department of Aeronautics and Astronautics, 2950 Hobson Way, Associate Fellow AIAA.

collect and analyze data collected by the WSN, make decisions based on the collected data, delivering only the most pertinent information to the operators or to conduct further action.

In this research, we address the challenge of optimal UAS path planning for collecting data from WSN. Additionally, with the data collected from the WSN, we enable the UAS the ability to modify its flight path to survey areas of higher interest, and delivering the most relevant information to the operators. This UAS-WSN architecture has the potential for a wide range of applications.

II. Motivation

Studies conducted at the Air Force Research Laboratory (AFRL) by Kingston¹, Marshall et al.², and Klesh et al.³ considered a UAS tasked to collect data from sparse but tactically placed ground sensors, making up a WSN. The engineers at AFRL envision a scenario where WSN are used to protect vital assets in remote austere locations, such as forward operating bases. Currently, flight path planning is done flying directly sensor to sensor, with no autonomous data analysis, or replanning capability. Point to point flight path planning often results in longer than necessary flight times and distances. A second concern is that the current architecture is not capable of autonomously determining and concentrating on areas within the WSN experiencing the highest ground activity. This is the main focus of this research.

We believe this is the first study of its kind, where the scenario posed is solved by the k -means clustering and direct collocation methods, using the data collected as the motivating factor for the flight path planning and re-planning.

III. Background

WSN collect, record, and report environmental disturbances within their immediate vicinity. Disturbances are recorded as electro-optical, doppler, seismic, magnetic or infrared information, and reported to overflying UAS¹. For example, passing vehicles may be recorded as seismic disturbances, time-stamped, and transmitted to the overflying UAS. WSN are currently employed in a variety of applications such as monitoring seabird habitats⁴, coral reefs and fisheries⁵, structural health⁶, forest fires⁷, fertilizer and pesticide applications in agriculture⁸, vehicular ground traffic in urban areas⁹, boarder patrols¹⁰, and tracking illegal boarder crossings[§].

Typical approaches for collecting WSN data include the use of ground based collection agents such as mobile robots.^{11,12} However if sensors are placed in hard to reach, rugged, or austere locations, the surface vehicle option quickly diminishes. Sensor nodes within communication range of each other often multi-hop the data back to a central collection sink. However, the inner most sensors, handling the highest volume of data traffic from the outer most sensors, often experience the highest rate of energy consumption.^{12,13} Since individual sensor units are considered low cost and disposable, they may have limited power reserves. The multi-hopping solution also requires sensors to be within line of sight of each other, limiting their geographical and spacial placement.

In some cases, individual WSN nodes are placed far enough apart they cannot communicate with each other. Furthermore, the sensors are low powered and cannot communicate with overhead satellites, making the UAS an ideal candidate to collect their stored data. However, employing a UAS has trade-offs. These include weather limitations, airspace and communication frequency management requirements, UAS logistics and maintenance, as well as processing and response time.¹

AFRL envisions a concept whereby a UAS visits all the sensors of a WSN to collect and relay sensor data to necessary users, obviating the ground-based central collection sink or the use of ground collection agents. This architecture enables a much larger WSN, free of line of sight communications requirements. AFRL proposes a scenario where the WSN collects and records the movement of vehicular and pedestrian traffic within its sensing range. The overflying UAS will collect sensor data, prioritize those with the highest activity, and re-compute an optimal flight path to increase the revisit rate areas of the WSN with the highest activity.

IV. Problem Statement

There are two main objectives of this research. First given a WSN and multiple UAS, develop a method of assigning clusters of sensor nodes to individual UAS. Second, develop an optimal control method to ensure sufficient data collection from all sensors within each cluster, in the least amount of time and control effort. Given initial data collected from the first orbits of the WSN, re-accomplish the first and second objectives by regrouping the sensors into

[§]Case study cited by Applied Research Associates on Jan 30, 2014 and can be found at <http://forcepro.ara.com/case-study-california-rancher-deploys-e-ugs-seismic-sensors-intercept-illegal-immigrants>, accessed on Apr 14, 2014

clusters experiencing the highest and lowest activity, and determine new optimal paths for these new groups. With the new sensor clusters, ensure increased revisit rates to high activity areas while maintaining or decreasing revisit rates to areas of low or no activity.

The second objective in the problem statement, develop an optimal control method to ensure sufficient data collection from all sensors, was addressed by Jodeh et al.¹⁴ Portions of that work are reproduced in this paper. The procedure for UAS to sensor assignment and sensor prioritization will be addressed in more detail in this paper.

V. Assumptions and Limitations

In order to address the problem statements, several assumptions and limitations must be made constrain the scope of the problem and keep it tractable.¹⁴

A. Sensor Assumptions and Limitations

The WSN topology is homogeneous. The transmission power, communication and sensing capability, and storage capacity are uniform across the WSN. Sensor locations are known to the UAS a priori and do not move once placed. The sensors are also assumed to have sufficient on board data storage and energy reserves to last the duration of an entire UAS mission. Sensors are equipped with omni-directional antennas and are limited to communicating with the UAS only.

B. UAS Assumptions and Limitations

The UAS is modeled as a simple point mass system, capable of climb, and to change heading and velocity. Wind and other external forces are neglected. Only single orbits of the WSN are calculated as subsequent orbits are considered identical until such time re-planning is called needed. The UAS size, weight, and performance specifications are limited a Tier N/A or Tier 1 type vehicle, as defined by the US Air Force¹⁵. The UAS is also assumed to have unlimited on-board data storage and range.

C. Problem Scope

Many challenges are associated with airborne aircraft collecting data from a network of ground sensors. To keep the problem tractable, this paper will not address path following (inner loop control) by the UAS on the commanded path. Nor will we tackle the challenge of WSN lifetime. Many authors were concerned with data latency as the collected information was required to be delivered back to a base station for analysis. This study presumes data is collected and prioritized on-board the UAS. Once the sensors with the most significant data are determined and overflowed, that data can be transmitted or delivered back to the user. Various communication methods, bandwidths, and protocols will not be addressed.

VI. Methodology

The problem is broken into several sub-problems. The sensor to UAS assignment is set up as a partition and clustering problem. The assignment algorithm is used for initial UAS-sensors assignments when multiple UAS available. It will also be used after the initial data collection orbits and sensor prioritization, to increase the revisit rates to high priority sensors. This information will be fed to the aforementioned assignments and path optimization routine. The UAS flight path is determined using an optimal control problem (OCP), solved by direct collocation methods, also known as pseudospectral methods.^{16,17}

A simplified flow chart in Figure 1 illustrates the proposed solution process. We take the flight test and research point of view, in other words, some processes won't be handled by the UAS necessarily. We envision this process running on-board the UAS in an operational setting, but for research purposes, we accept that the solutions are calculated on a ground station and uploaded to the UAS. Once the initial information is input, the assignment algorithm and optimal path planner compute their respective solutions. During testing the optimal paths can then be uploaded to the UAS from a ground-based computer. Once an initial orbit of the WSN is completed, the data can either be sent back or processed on board. The UAS continues to repeat the original flight path until a new one is calculated. For research purposes, we assume only a few contiguous sensors experience a higher than average volume of ground activity, thus demanding more attention from a particular UAS. New optimal paths are calculated based on information from the initial orbit. These newer paths may add or subtract sensors from individual UAS responsibilities, and enable a particular UAS to concentrate on the areas of the WSN with the highest activity.

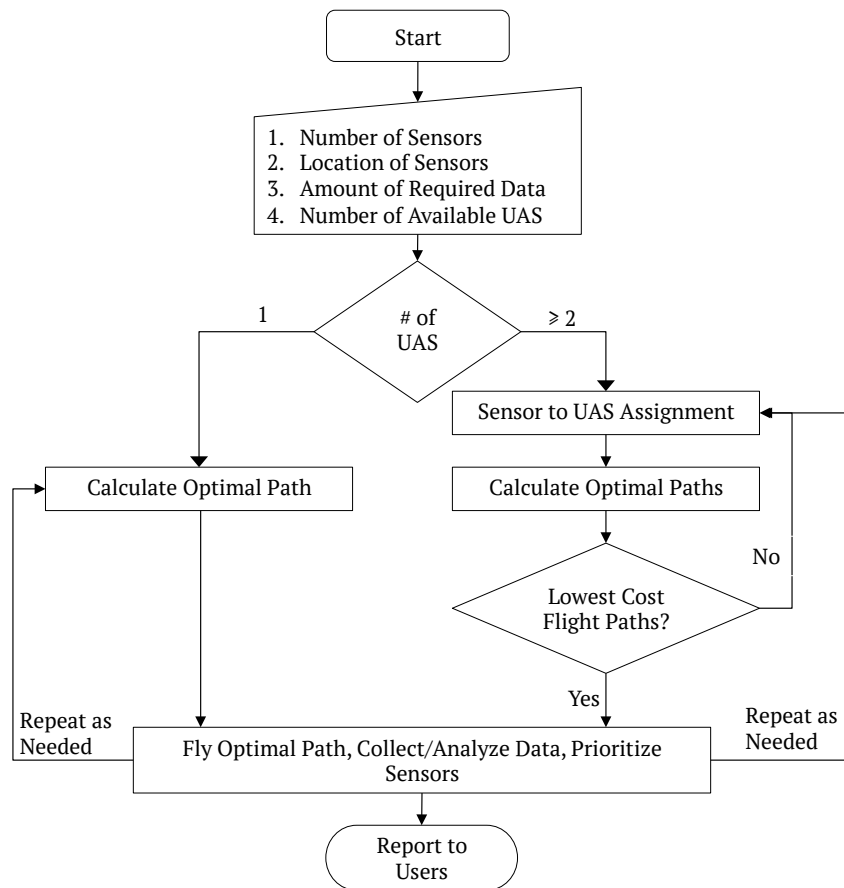


Figure 1. Flow Chart for UAS Data Collection in WSN

A. Sensor to UAS Assignment

Given a set of k UAS, the individual sensors can be grouped into k clusters. Clustering takes advantage of the available UAS and reduces the time required to survey all the sensors. The k -means algorithm is based on work by Lloyd¹⁸ and Arthur and Vassilvitskii¹⁹. The Arthur and Vassilvitskii routine implemented via MATLAB routine, is used here. k -means was chosen because it treats each of the sensors as a location in space, and partitions the clusters based on some chosen distance measure.¹⁹ Each of the clusters uses a centroid, where the sum of all distances from points, i.e. sensors, to the centroid is minimized. The number of clusters equals the number of available UAS. By trading points in-between the clusters, or moving the centroid around, the sum of the individual sensor to centroid distances changes. Once the lowest sum is reached, the algorithm is satisfied. The sensors are assigned to particular clusters, and clusters assigned to individual UAS. Fly/No-Fly zones are then established to ensure optimal paths are deconflicted for multi-UAS operations.

To keep a particular UAS in the Fly/No-Fly zone, the position of the UAS must be less than the zone's radius. Conversely, to keep a particular UAS outside of the Fly/No-Fly zone, the position of the UAS must be greater than the zone's radius. Allowing any of the UAS to fly along the radius is not acceptable as that would introduce conflicting flight paths.

B. Optimal Flight Path

Once the sensor clusters have been determined, individual optimal flight paths are calculated for each UAS. An OCP is formulated to find the control resulting in an optimal flight path. Specifically, find the optimal control \mathbf{u}^* , in the set of admissible controls, U , that minimizes the following cost functional. This specific OCP first appeared and updated in Jodeh^{14,20} and is reproduced here for completeness with some added detail. The OCP presented here introduces the Fly/No-Fly zone and multiple-UAS formulation.

The cost functional is comprised of the terminal cost, the total flight time, t_f , and the running costs, pitch rate control, $\dot{\theta}$, heading rate control, $\dot{\psi}$, and velocity rate control, \dot{v} . Individual time and control weights α and β are used to scale and tune the optimal trajectory. The sum of these costs are minimized in the OCP, in Equation 1

$$\min_{\mathbf{u} \in U} J = \alpha(t_{1f} + \dots + t_{kf}) + \int_{t_0}^{t_f} \beta_1(\dot{v}_1^2 + \dots + \dot{v}_k^2) + \beta_2(\dot{\theta}_1^2 + \dots + \dot{\theta}_k^2) + \beta_3(\dot{\psi}_1^2 + \dots + \dot{\psi}_k^2) dt \quad (1)$$

for $k = 1, 2, \dots, K$ where K =Number of UASs

The dynamic constraints are the aircraft equations of motion in Equations 2. The cost functional is subject to these constraints.

$$\dot{x} = v \cos \theta \cos \psi \quad (2a)$$

$$\dot{y} = v \cos \theta \sin \psi \quad (2b)$$

$$\dot{z} = v \sin \theta \quad (2c)$$

The states of the problem include UAS position along the x, y , and z coordinate frames, the pitch angle θ , heading angle ψ and the velocity v . The controls are pitch rate $\dot{\theta}$, heading rate $\dot{\psi}$, and acceleration \dot{v} .

$$\mathbf{x} = [x, y, z, \theta, \psi, v]^T \quad (3a)$$

$$\mathbf{u} = [\dot{\theta}, \dot{\psi}, \dot{v}]^T \quad (3b)$$

The boundary conditions are given by Equations 4.

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (4a)$$

$$\mathbf{x}(t_f) = \mathbf{x}_f \quad (4b)$$

The states and controls are limited by the path constraints in Equations 5a.

$$\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \quad (5a)$$

$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max} \quad (5b)$$

$$\|\hat{\mathbf{x}}_k(t) - \tilde{\mathbf{x}}_n\|_2 > r_n \text{ for } k \neq n \quad (5c)$$

$$\|\hat{\mathbf{x}}_k(t) - \tilde{\mathbf{x}}_n\|_2 < r_n \text{ for } k = n \quad (5d)$$

Finally, the integral constraints act as the motivating factor of the optimal path. Equation 6 states that integral of the data rate C of sensor j over the flight time must be greater than or equal to some minimum required data, DP . The cost functional and dynamics are subject to the integral constraint in Equation 6

$$\int_{t_0}^{t_f} C_j dt \geq DP_j \quad \forall j. \quad (6)$$

Jodeh et al. determined the communication model based on empirical data from flight test in 2015. The reader is directed to that research for detailed information on the communications model. The maximum data rate is 5 kbit/s up to a distance of 1 km. The sensor data rates are calculated in Equation 7 as a function of the distance d between the UAS and each sensor node. The distance calculation is given in Equation 8.

$$C_j = \frac{C_{max}}{1 + e^{-s(D_{max}-d_j)}} \quad (7)$$

$$d_j = \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad \forall j \quad (8)$$

Direct Collocation (Pseudospectral) Methods have become increasingly popular to directly solve for optimal control problems solutions²¹⁻²³. Numerous publications in the last decade have thoroughly described and developed direct collocation methods^{16,24}. In the areas of trajectory optimization, the reader is directed to^{20,25-28}.

The MATLAB-based, General Purpose Optimal Control Software-II (GPOPS-II) was the program used to solve for the optimal flight path. GPOPS-II directly solves the continuous time optimal control problem.²⁹ GPOPS-II transcribes the continuous-time optimal control problem into a discrete finite-dimensional static optimization problem that is solved as a Non-Linear Programming (NLP) problem. GPOPS-II is implemented using the *hp*-adaptive version of the Radau pseudospectral method. This particular method is an orthogonal collocation, Gaussian quadrature, implicit integration method in which the collocation is performed at the Legendre-Gauss-Radau quadrature points.³⁰ An adaptive mesh refinement method is implemented that determines the number of mesh intervals and the degree of the approximating polynomial for each mesh interval in order to satisfy the specified error tolerance.³⁰ GPOPS-II can be set to prioritize finding the degree of the approximating polynomial before refining the mesh interval or vice versa.

VII. Results

The results section illustrates the solution for single and multi-UAS scenarios. The scenarios chosen are relatively simple, resulting in intuitive solutions. This approach helps to verify functionality. More complex scenarios are considered for future work. Both of the following scenarios start with simple, 16 sensor WSN laid in a uniform, equidistant, square grid. Sensors observing the increased activity are manually selected for these examples. Also, for both scenarios, the UAS same altitude as the sensors, rendering the problem two-dimensional. The time and control weights were determined by Jodeh et al.¹⁴ as $\alpha=0.01$ and $\beta=[1, 1, 0.001]$.

A. Single UAS Scenario

Given one available UAS and the 16 sensor WSN, clustering and assignment are not required. The UAS is required to survey the entire WSN. The initial lay out is show in Figure 2(a). Reference the legend in Figure 2(b) for Figures 2(a), 3(a), and 4(a).

Next, the UAS was tasked with collecting 10 kbit from each of the sensors. The optimal flight path is illustrated Figure 3(a). Three concentric rings indicate 99%, 50%, and 1% value of the total data rate as a function of the distance from the center of the sensor, discussed further in Jodeh et al.¹⁴ This information is included to help the reader visualize when the UAS is connected to each of the sensors and gives insight to the shape of the optimal path. After the first orbit, the square shaped sensors indicate areas with the highest activity. Figure 3(b) illustrates the pitch, heading, and velocity states of the UAS. The velocity is held constant at 30 m/s for the duration of the initial orbit, which lasted 799 seconds. The optimizer confirmed the UAS successfully collected 10 kbit from each sensor.

The amount of data required from the high activity sensors is arbitrarily set at 1 Mbit, 100 times the amount of the low activity sensors. Once the high activity sensors were identified, a new optimal path was calculated to ensure the UAS concentrated more flight time and collected more data from them. The recomputed optimal flight path is shown in Figure 4(a). The new flight path clearly shows the UAS inside the highest data rate region of the four sensors of interest. However, flight path alone does not tell the whole story. In Figure 4(b), the velocity slows from

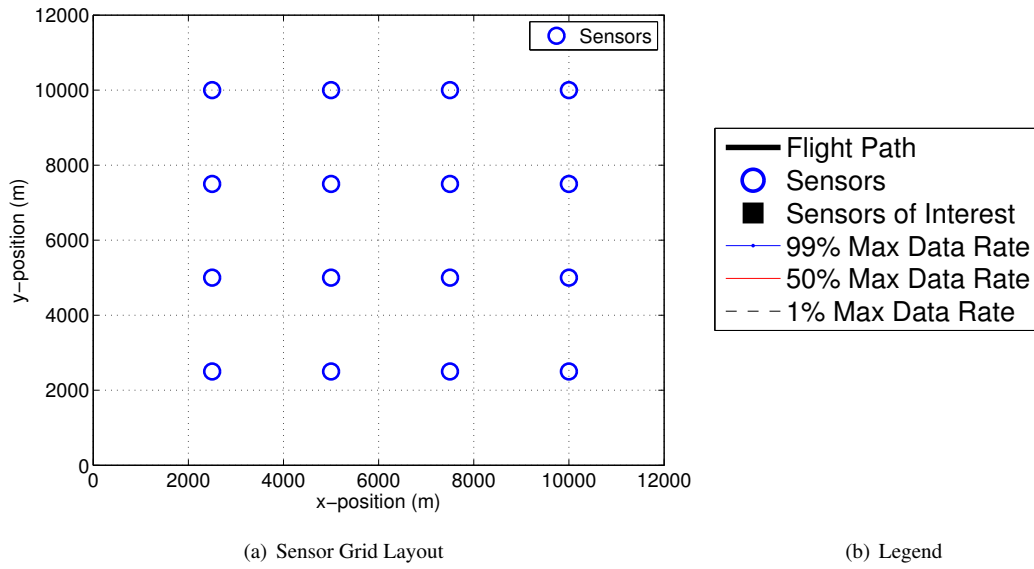


Figure 2. Simple 16 sensor layout example for single UAS scenario. Reference above legend for Figures 3(a) and 4(a).

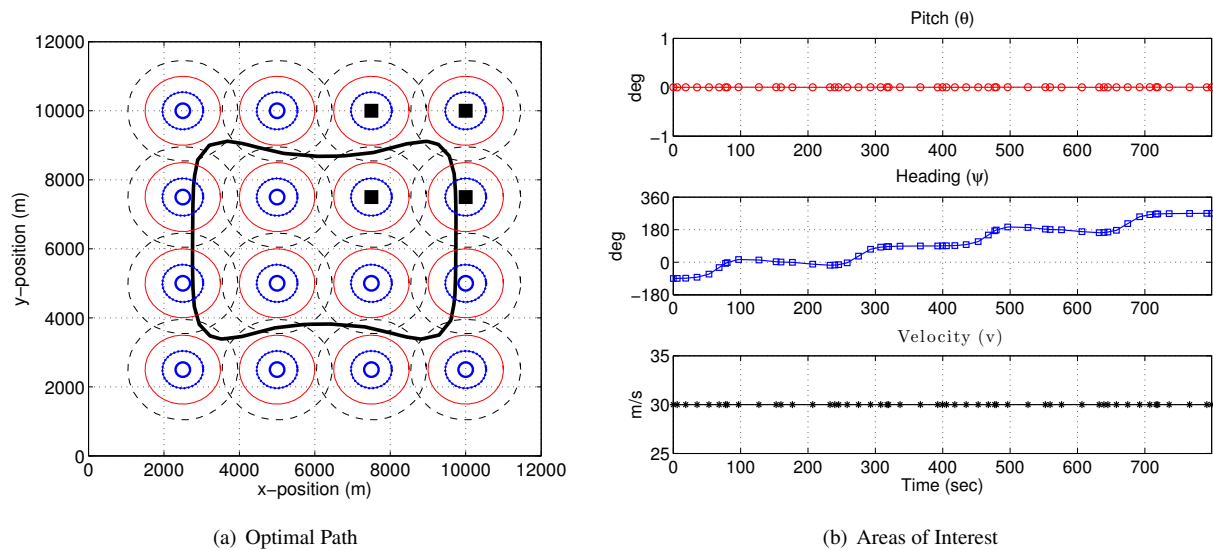


Figure 3. Figure 3(a) is the optimal flight path for the initial sensor layout given one available UAS. The square markers identify sensors with high activity after the initial orbit. Figure 3(b) illustrates the states of the UAS on initial orbit.

Figure 1 shows a 4x4 grid of 16 circular regions. Each region contains concentric circles (red and blue) and a dashed black circle. A thick black line traces a path through the grid, connecting specific points marked with black squares. The axes are labeled 'x-position (m)' and 'y-position (m)' ranging from 0 to 12000.

The figure consists of three vertically stacked plots sharing a common x-axis representing Time (sec) from 0 to 1400.

- Top Plot:** Pitch (θ) in degrees (deg) vs. Time (sec). The y-axis ranges from -1 to 1. The data points (red circles) are clustered around 0 degrees, indicating a constant pitch.
- Middle Plot:** Heading (ψ) in degrees (deg) vs. Time (sec). The y-axis ranges from -180 to 360. The data points (blue squares) show a fluctuating heading, starting around -150 degrees, peaking near 180 degrees around 1100 seconds, and ending near 180 degrees.
- Bottom Plot:** Velocity (v) in m/s vs. Time (sec). The y-axis ranges from 0 to 30. The data points (black circles) show a velocity profile that starts at 30 m/s, drops to 10 m/s around 500 seconds, and then exhibits several sharp peaks and troughs between 10 and 30 m/s.

Figure 4. Re-planned optimal paths for two UAS given updated sensor areas of high activity.

In the second scenario, two UAS will fly the same 16 sensor WSN. Given the two UAS, the clustering algorithm splits the WSN into two clusters of eight sensors respectively. The clusters and their respective centroids are plotted in Figure 5(a). The centroid helps to identify clusters for the optimization routine and also provide a visual cue for the reader. This is especially important in a more randomly scattered WSN. For the duration of the scenario, the minimum data required is set at 10 kbit. Increasing the amount of data required is not required as a single UAS will be devoted to the high interest sensors and capable of collecting any amount of required data. Reference the legend in Figure 5(b) for Figures 5, 6, and 7.

Finally, with the sensors of interest identified, the clustering algorithm regroups the WSN into clusters of high and low activity. As in the single UAS scenario, the sensors of interest are denoted by square markers on the plot. Unlike the single UAS scenario, this is also now a new cluster, with its own assigned UAS. Transit between the initial and re-planned flight paths by the UAS are assumed to be negligible and that the UAS are deconflicted by the operator, altitude, or in time to reach their new orbits.

In both scenarios, the minimum amount of required data was collected. In the second scenario, the high activity sensors received constant monitoring from a dedicated UAS after they were identified in the initial orbits. The results

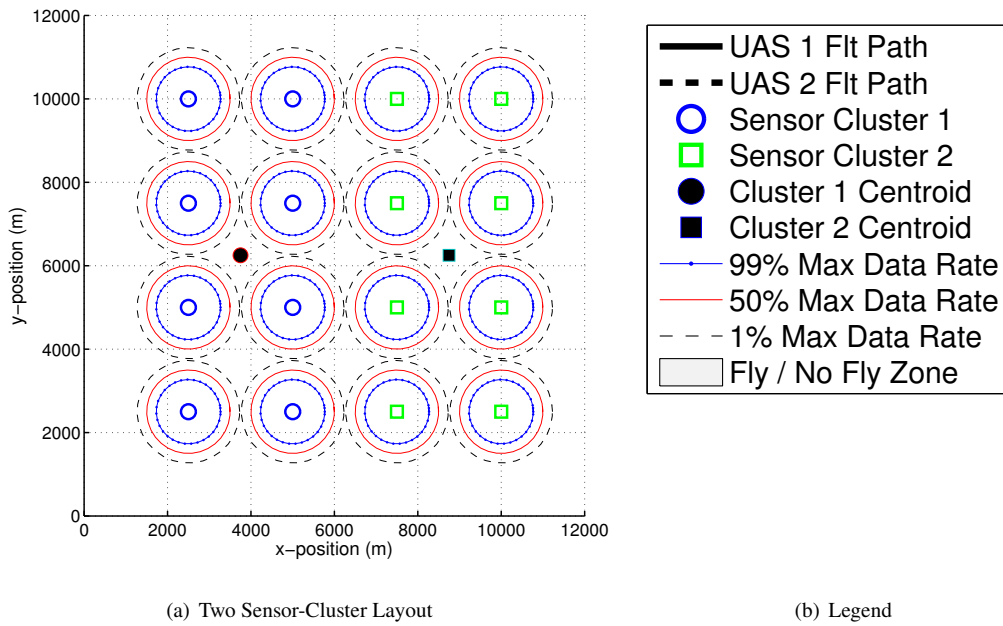


Figure 5. Simple 16 sensor, two cluster layout example for two UAS scenario. Reference above legend for Figures 6 and 7.

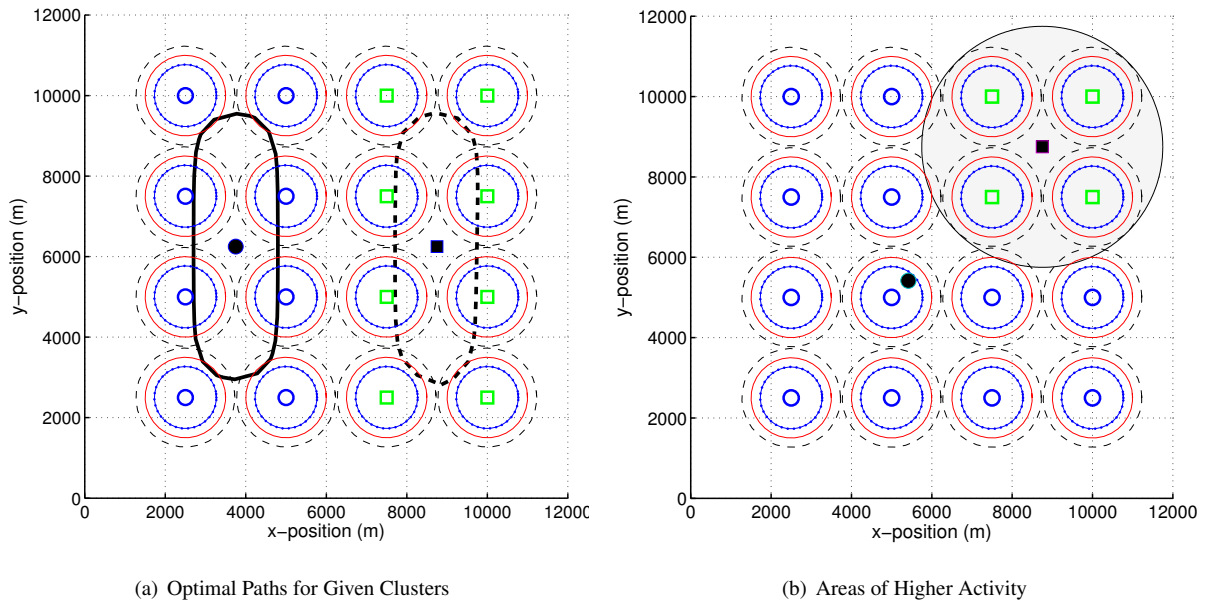


Figure 6. Figure 6(a) are the optimal flight paths for the initial sensor layout given two available UAS. Figure 6(b) indicates areas of higher activity as detected by the initial UAS orbits.

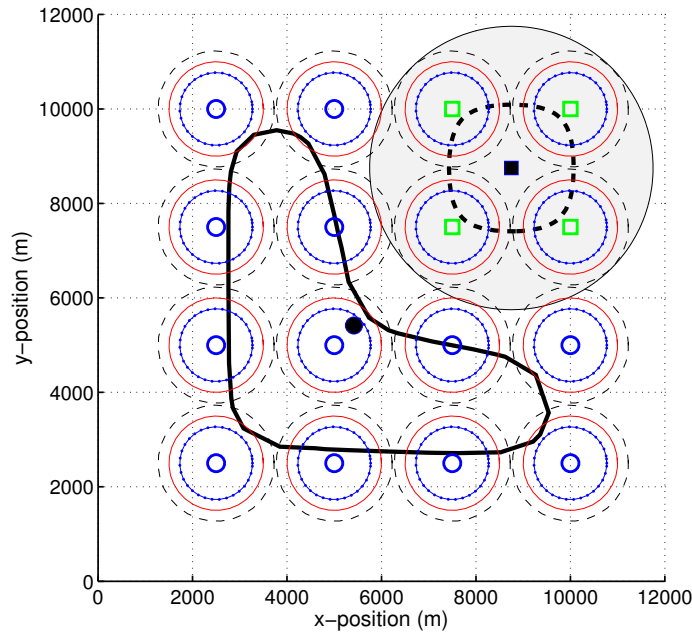


Figure 7. Re-planned optimal paths for two UAS given updated sensor areas of high activity (square markers). Fly / No-Fly zone used to deconflict assigned airspace, ensuring optimal paths do not intersect.

all satisfied the OCP constraints and the error tolerances set in GPOPS-II and the NLP. Although these earlier results are promising, many questions still remain unanswered. For example, in multi-UAS scenarios, can UAS be prevented from detecting or communicating with sensors in neighboring clusters. To deconflict flight paths, can multiple UAS operate at various altitudes and still communicate with sensors on the ground? Also the case of non-contiguous high activity sensors remains to be tested. Flight tests planned for spring 2015 will help to validate the optimal paths and confirm the data collection assumptions.

IX. Conclusion

The solution method presented lays the ground work for single and multiple UAS sensor assignment, path planning, and data collection in WSN. Initial results verify the method as a viable alternative to ground based collection agents or to multi-hopping data. However, further testing of the algorithms are required, including answering the questions detailed in the previous section.

Acknowledgement

The authors would like to especially recognize and thank Dr Derek Kingston, Mr Steve Rasmussen, and Lt Riley Livermore of the Air Force Research Laboratories, Control Science Center of Excellence, who graciously provided us with the challenging problem concept, background information, and endless support.

References

- ¹Kingston, D., "Intruder Tracking Using UAV Teams and Ground Sensor Networks," Tech. rep., Control Science Center of Excellence, Air Force Research Labs, Wright Patterson AFB, OH, USA, 2011.
- ²Marshall, C., Mears, M., Kingston, D., and Rasmussen, S., "2010 ICE-T Cooperative Control Flight Testing," *Infotech@ Aerospace 2011*, No. AIAA 2011-1486.
- ³Klesh, A. T., Kabamba, P. T., and Girard, A. R., "Path Planning for Cooperative Time-Optimal Information Collection," *2008 American Control Conference*, 2008.
- ⁴Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J., "Wireless sensor networks for habitat monitoring," *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ACM, 2002, pp. 88–97.
- ⁵Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., and Corke, P., "Data collection, storage, and retrieval with an underwater sensor network," *Proceedings of the 3rd international conference on Embedded networked sensor systems*, ACM, 2005, pp. 154–165.

- ⁶Mascareñas, D., Flynn, E., Todd, M., Park, G., and Farrar, C., “Wireless sensor technologies for monitoring civil structures,” *Sound and Vibration*, Vol. 42, No. 4, 2008, pp. 16–21.
- ⁷Aslan, Y. E., Korpeoglu, I., and Ulusoy, Ö., “A framework for use of wireless sensor networks in forest fire detection and monitoring,” *Computers, Environment and Urban Systems*, Vol. 36, No. 6, 2012, pp. 614–625.
- ⁸Façal, B. S., Costa, F. G., Pessin, G., Ueyama, J., Vieira, H. F., Colombo, A., Fini, P. H., Villas, L., Osório, F. S., Vargas, P. A., et al., “The Use of Unmanned Aerial Vehicles and Wireless Sensor Networks for Spraying Pesticides,” *Journal of Systems Architecture*, 2014.
- ⁹Xiao, Q., Xiao, B., Luo, J., and Liu, G., “Reliable navigation of mobile sensors in wireless sensor networks without localization service,” *Quality of Service, 2009. IWQoS. 17th International Workshop on*, IEEE, 2009, pp. 1–9.
- ¹⁰Sun, Z., Wang, P., Vuran, M. C., Al-Rodhaan, M. A., Al-Dhelaan, A. M., and Akyildiz, I. F., “BorderSense: Border patrol through advanced wireless sensor networks,” *Ad Hoc Networks*, Vol. 9, No. 3, 2011, pp. 468–477.
- ¹¹Almi’ani, K., Viglas, A., and Libman, L., “Tour and Path Planning Methods for Efficient Data Gathering using Mobile Elements,” *International Journal of Ad Hoc and Ubiquitous Computing*, accepted for publication, 2014.
- ¹²Castañó, F., Rossi, A., Sevaux, M., and Velasco, N., “On the use of multiple sinks to extend the lifetime in connected wireless sensor networks,” *Electronic Notes in Discrete Mathematics*, Vol. 41, 2013, pp. 77–84.
- ¹³Gu, Y., Zhao, B.-H., Ji, Y.-S., and Li, J., “Theoretical treatment of target coverage in wireless sensor networks,” *Journal of Computer Science and Technology*, Vol. 26, No. 1, 2011, pp. 117–129.
- ¹⁴Jodeh, N. M., Cobb, R. G., and Livermore, R. S., “Optimal Airborne Trajectories for Data Collection from Wireless Sensor Networks by Direct Collocation Methods,” AIAA, AIAA Guidance, Navigation and Control Conference, January 2015.
- ¹⁵“Unmanned Systems Integrated Roadmap FY 2011-2036,” No. ADA558615, Office of the Under Secretary of Defense (Acquisition Technology and Logistics), Washington DC, Oct 2011.
- ¹⁶Huntington, G. T., *Advancement and analysis of a Gauss pseudospectral transcription for optimal control problems*, Ph.D. thesis, Citeseer, 2007.
- ¹⁷Rao, A. V., “A survey of numerical methods for optimal control,” .
- ¹⁸Lloyd, S., “Least squares quantization in PCM,” *Information Theory, IEEE Transactions on*, Vol. 28, No. 2, 1982, pp. 129–137.
- ¹⁹Arthur, D. and Vassilvitskii, S., “k-means++: The advantages of careful seeding,” *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
- ²⁰Jodeh, N. M., Coon, T. E., Masternak, T. J., Cobb, R. G., and Agte, J. S., “Optimal Airborne Trajectories for Data Collection from Emplaced Ground Sensor Arrays,” January 2014.
- ²¹Ross, S. M., Cobb, R. G., Baker, W. P., and Harmon, F. G., “Implementation lessons and pitfalls for real-time optimal control with stochastic systems,” *Optimal Control Applications and Methods*, 2014.
- ²²Gauntt, R. D., *Aircraft Course Optimization Tool Using GPOPS MATLAB Code*, No. afit/gse/env/12-m03, Air Force Institute of Technology, Wright Patterson AFB, Dayton, OH, March 2012.
- ²³Jorris, T. R. and Cobb, R. G., “Multiple method 2-D trajectory optimization satisfying waypoints and no-fly zone constraints,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 543–553.
- ²⁴Benson, D., *A Gauss pseudospectral transcription for optimal control*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.
- ²⁵Suplisson, A., *Automatic Collision Avoidance for Larger Air Vehicles*, Ph.D. thesis, AFIT, 2014.
- ²⁶Masternak, T., *Hypersonic Path Planning with Direct Collocation Methods*, Ph.D. thesis, AFIT, 2014.
- ²⁷Smith, N. E., Cobb, R. G., Pierce, S. J., and Raska, V. M., “Optimal Collision Avoidance Trajectories via Direct Orthogonal Collocation for Unmanned/Remotely Piloted Aircraft Sense and Avoid Operations,” 2014.
- ²⁸Geiger, B. R., Horn, J. F., Sinsley, G. L., Ross, J. A., Long, L. N., and Niessner, A. F., “Flight testing a real-time direct collocation path planner,” *Journal of guidance, control, and dynamics*, Vol. 31, No. 6, 2008, pp. 1575–1586.
- ²⁹Patterson, M. A., R. A. V., “GPOPS-II: A MATLAB Software for Solving Multi-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Method and Sparse Nonlinear Programming,” *ACM Transactions on Math. Software*, Vol. 39, No. 3, Jul 13.
- ³⁰Patterson, M. A., R. A. V., *GPOPS-II Version 1.0: A General-Purpose MATLAB Toolbox for Solving Optimal Control Problems Using the Radau Pseudospectral Method*, University of Florida, Gainesville, FL 32611-6250, USA, 1st ed., January 2013.

Bibliography

- [1] *DoD Unmanned Systems Integrated Roadmap*. US Department of Defense, 2013.
- [2] Gregory J Pottie. Wireless sensor networks. In *Information Theory Workshop, 1998*, pages 139–140. IEEE, 1998.
- [3] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.
- [4] Wenrui Zhao and Mostafa H Ammar. Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *Distributed Computing Systems, 2003. FTDCS 2003. Proceedings. The Ninth IEEE Workshop on Future Trends of*, pages 308–314. IEEE, 2003.
- [5] Lang Tong, Qing Zhao, and Srihari Adireddy. Sensor networks with mobile agents. In *Military Communications Conference, 2003. MILCOM'03. 2003 IEEE*, volume 1, pages 688–693. IEEE, 2003.
- [6] Kevin L. Moore, Michael J. White, Robert J. Bamberger, and David P. Watson. Data exfiltration from unattended ground sensors using cooperating uavs. *Proc. SPIE*, 5804:279–290, 2005.
- [7] Chris Fraley and Adrian E Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588, 1998.
- [8] PB Sujit and Randy Beard. Multiple mav task allocation using distributed auctions. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 20–23, 2007.
- [9] Luke B Johnson, Sameera S Ponda, Han-Lim Choi, and Jonathan P How. Asynchronous decentralized task allocation for dynamic environments. In *Proceedings of the AIAA Infotech@ Aerospace Conference, St. Louis, MO*, 2011.
- [10] Luca F Bertuccelli, Han-Lim Choi, Peter Cho, and Jonathan P How. Real-time multi-uav task assignment in dynamic and uncertain environments. In *presentado al AIAA Guidance, Navigation, and Control Conference, Chicago, Illinois*, 2009.
- [11] Tim Bakker and Robert H Klenke. Dynamic multi-task allocation for collaborative unmanned aircraft systems. In *52nd Aerospace Sciences Meeting, AIAA-SciTech 2014*. AIAA, 2014.
- [12] Xu Li, Amiya Nayak, and Ivan Stojmenovic. Sink mobility in wireless sensor networks. *Wireless Sensor and Actuator Networks*, page 153, 2010.

- [13] D. Kingston. Intruder tracking using uav teams and ground sensor networks. Technical report, Control Science Center of Excellence, Air Force Research Labs, Wright Patterson AFB, OH, USA, 2011.
- [14] Chad Marshall, Mark Mears, Derek Kingston, and Steve Rasmussen. 2010 ice-t cooperative control flight testing. In *Infotech@ Aerospace 2011*, number AIAA 2011-1486.
- [15] Andrew T. Klesh, Pierre T. Kabamba, and Anouck R. Girard. Path planning for cooperative time-optimal information collection. In *2008 American Control Conference*, 2008.
- [16] *Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap*. Wiley & Sons, Inc., Washington D.C., 2013.
- [17] *Department of Defense Dictionary of Military and Associated Terms*. Joint Publication 1-02. Department of Defense, January 2014.
- [18] Khaled Almi'ani, Anastasios Viglas, and Lavy Libman. Tour and path planning methods for efficient data gathering using mobile elements. *International Journal of Ad Hoc and Ubiquitous Computing*, accepted for publication, 2014.
- [19] R Moazzez-Estanjini and I Ch Paschalidis. On delay-minimized data harvesting with mobile elements in wireless sensor networks. *Ad Hoc Networks*, 10(7):1191–1203, 2012.
- [20] Yaoyao Gu, Doruk Bozdağ, Robert W. Brewer, and Eylem Ekici. Data harvesting with mobile elements in wireless sensor networks. *Computer Networks*, 50(17):3449 – 3465, 2006.
- [21] Adrien Jimenez-Gonzalez, Jose Ramiro Martinez-de Dios, and Anebal Ollero. An integrated testbed for cooperative perception with heterogeneous mobile and static sensors. *Sensors*, 11(12):11516–11543, 2011.
- [22] Onur Tekdas, Volkan Isler, Jong Hyun Lim, and Andreas Terzis. Using mobile robots to harvest data from sensor fields. *IEEE Wireless Communications*, 16(1):22, 2009.
- [23] Karthik Dantu, Mohammad Rahimi, Hardik Shah, Sandeep Babel, Amit Dhariwal, and Gaurav S Sukhatme. Robomote: enabling mobility in sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 55. IEEE Press, 2005.
- [24] Ben Pearre and Timothy X Brown. Energy conservation in sensor network data ferrying: A reinforcement metalearning approach. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 79–85. IEEE, 2012.

- [25] Anthony J Carfang, Eric W Frew, and Derek B Kingston. A cascaded approach to optimize aircraft trajectories for persistent data ferrying. *Journal of Guidance, Control, and Dynamics*, 2013.
- [26] Yung-Liang Lai and Jehn-Ruey Jiang. A genetic algorithm for data mule path planning in wireless sensor networks. *Applied Mathematics & Information Sciences*, 7(1), 2013.
- [27] Steven Rasmussen and Derek Kingston. Development and flight test of an area monitoring system using unmanned aerial vehicles and unattended ground sensors. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 1215–1224. IEEE, 2015.
- [28] Nidal M. Jodeh, Timothy Coon, Tadeusz J. Masternak, Richard Cobb, and Jeremy S. Agte. *Optimal Airborne Trajectories for Data Collected from Emplaced Ground Sensor Arrays*. American Institute of Aeronautics and Astronautics, January 2014.
- [29] Amir Ajorlou, Abdollah Homaifar, Albert Esterline, Jay Moore, and Robert Bamberger. Robust multi-uav data collection. (AIAA 2007-2811), 2007.
- [30] Deepak Bhadauria, Onur Tekdas, and Volkan Isler. Robotic data mules for collecting data over sparse sensor fields. *Journal of Field Robotics*, 28(3):388–404, 2011.
- [31] Jason Lester Hill. *System architecture for wireless sensor networks*. PhD thesis, University of California, Berkeley, 2003.
- [32] Weifa Liang, Jun Luo, and Xu Xu. Prolonging network lifetime via a controlled mobile sink in wireless sensor networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6. IEEE, 2010.
- [33] Can Tunca, Sinan Isik, M Yunus Donmez, and Cem Ersoy. Distributed mobile sink routing for wireless sensor networks: A survey. *Communications Surveys & Tutorials, IEEE*, 16(2):877–897, 2014.
- [34] Ming Ma and Yuanyuan Yang. Sencar: an energy-efficient data gathering mechanism for large-scale multihop sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 18(10):1476–1488, 2007.
- [35] Douglas Guimarães Macharet, Armando Alves Neto, Vilar Fiuza da Camara Neto, and Mario Fernando Montenegro Campos. Data gathering tour optimization for dubins’ vehicles. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.
- [36] Anthony Carfang, Eric W Frew, and Timothy X Brown. Improved delay-tolerant communication by considering radio propagation in planning data ferry navigation. In *AIAA Guidance, Navigation, and Control Conference*, 2010.

- [49] Forecasts, Office of Aviation Policy Performance Analysis Division, and Plans. *FAA Aerospace Forecasts FY 2014-2034*. Federal Aviation Administration, 2014.
- [50] Peng Wei, Quanquan Gu, and Dengfeng Sun. Wireless sensor network data collection by connected cooperative uavs. In *American Control Conference (ACC), 2013*, pages 5911–5916. IEEE, 2013.
- [51] Yunus Emre Aslan, Ibrahim Korpeoglu, and Özgür Ulusoy. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Computers, Environment and Urban Systems*, 36(6):614–625, 2012.
- [52] Fabian Castaño, André Rossi, Marc Sevaux, and N Velasco. On the use of multiple sinks to extend the lifetime in connected wireless sensor networks. *Electronic Notes in Discrete Mathematics*, 41:77–84, 2013.
- [53] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002.
- [54] Iuliu Vasilescu, Keith Kotay, Daniela Rus, Matthew Dunbabin, and Peter Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 154–165. ACM, 2005.
- [55] David Mascareñas, Eric Flynn, Michael Todd, Gyuhae Park, and Charles Farrar. Wireless sensor technologies for monitoring civil structures. *Sound and Vibration*, 42(4):16–21, 2008.
- [56] Bruno S Façal, Fausto G Costa, Gustavo Pessin, Jó Ueyama, Heitor F Vieira, Alexandre Colombo, Pedro H Fini, Leandro Villas, Fernando S Osório, Patrícia A Vargas, et al. The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides. *Journal of Systems Architecture*, 2014.
- [57] Qingjun Xiao, Bin Xiao, Jiaqing Luo, and Guobin Liu. Reliable navigation of mobile sensors in wireless sensor networks without localization service. In *Quality of Service, 2009. IWQoS. 17th International Workshop on*, pages 1–9. IEEE, 2009.
- [58] Zhi Sun, Pu Wang, Mehmet C Vuran, Mznah A Al-Rodhaan, Abdullah M Al-Dhelaan, and Ian F Akyildiz. Bordersense: Border patrol through advanced wireless sensor networks. *Ad Hoc Networks*, 9(3):468–477, 2011.
- [59] Derek B Kingston, Steven J Rasmussen, and Mark J Mears. Base defense using a task assignment framework. In *AIAA Guidance, Navigation, and Control Conference*, 2009.

- [60] James Spillings, Peter Burke, Antonios Tsourdos, Peter Silson, and Brian White. An evaluation of sensor and data fusion technologies for application within an integrated base defence system. 2009.
- [61] David Jea, Arun Somasundara, and Mani Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Distributed Computing in Sensor Systems*, pages 244–257. Springer, 2005.
- [62] Arun A Somasundara, Aditya Ramamoorthy, and Mani B Srivastava. Mobile element scheduling with dynamic deadlines. *Mobile Computing, IEEE Transactions on*, 6(4):395–410, 2007.
- [63] Songtao Guo, Xiaojian Wang, and Yuanyuan Yang. Topology control for maximizing network lifetime in wireless sensor networks with mobile sink. In *INFOCOM, 2013 Proceedings IEEE*, pages 240–248. IEEE, 2013.
- [64] Arun A Somasundara, Aditya Ramamoorthy, and Mani B Srivastava. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International*, pages 296–305. IEEE, 2004.
- [65] Yu Gu, Bao-Hua Zhao, Yu-Sheng Ji, and Jie Li. Theoretical treatment of target coverage in wireless sensor networks. *Journal of Computer Science and Technology*, 26(1):117–129, 2011.
- [66] Mohammed Abdulaal, Mohammed Algarni, Atif Shamim, and Christian Claudel. Unmanned aerial vehicle based flash flood monitoring using lagrangian trackers.
- [67] Jonathan Las Fargeas, Pierre Kabamba, and Anouck Girard. Cooperative surveillance and pursuit using unmanned aerial vehicles and unattended ground sensors. *Sensors*, 15(1):1365–1388, 2015.
- [68] Guoliang Xing, Tian Wang, Weijia Jia, and Minming Li. Rendezvous design algorithms for wireless sensor networks with a mobile base station. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 231–240. ACM, 2008.
- [69] Miao Zhao and Yuanyuan Yang. Optimization-based distributed algorithms for mobile data gathering in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 11(10):1464–1477, 2012.
- [70] Almi’ani K. A.alqaralleh, B. Mobile elements scheduling for periodic sensor applications. *International Journal of Wireless & Mobile Networks*, 6(1):15–30, February 2014.

- [71] Ko-Ming Chiu and Jing-Sin Liu. Path planning of a data mule for data collection in the sensor network by using an improved clustering-based genetic algorithm. In *International Conference on Information Technology*, volume 10, 2012.
- [72] Dac-Tu Ho and Shigeru Shimamoto. Highly reliable communication protocol for wsn-uav system employing tdma and pfs scheme. In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 1320–1324. IEEE, 2011.
- [73] Andrea Giorgetti, Matteo Lucchi, Marco Chiani, and Moe Z Win. Throughput per pass for data aggregation from a wireless sensor network via a uav. *Aerospace and Electronic Systems, IEEE Transactions on*, 47(4):2610–2626, 2011.
- [74] Jian Ouyang, Yi Zhuang, Min Lin, and Jia Liu. Optimization of beamforming and path planning for uav-assisted wireless relay networks. *Chinese Journal of Aeronautics*, 27(2):313–320, 2014.
- [75] Feng Jiang and A Lee Swindlehurst. Optimization of uav heading for the ground-to-air uplink. *Selected Areas in Communications, IEEE Journal on*, 30(5):993–1005, 2012.
- [76] Johnson Kuruvila, Amiya Nayak, and Ivan Stojmenovic. Hop count optimal position-based packet routing algorithms for ad hoc wireless networks with a realistic physical layer. *Selected Areas in Communications, IEEE Journal on*, 23(6):1267–1275, 2005.
- [77] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.
- [78] Christos H Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. In *Automata, Languages and Programming*, pages 610–620. Springer, 1989.
- [79] William R Stewart Jr. Chinese postman problem. In *Encyclopedia of Operations Research and Management Science*, pages 161–164. Springer, 2013.
- [80] Khaled Elbassioni, Aleksei V Fishkin, Nabil H Mustafa, and René Sitters. Approximation algorithms for euclidean group tsp. In *Automata, Languages and Programming*, pages 1115–1126. Springer, 2005.
- [81] Sudarsanan Nesamony, Madhan Karky Vairamuthu, Maria Orlowska, and Shazia Sadiq. On optimal route computation of mobile sink in a wireless sensor network. Technical report, 2006.
- [82] Sudarsanan Nesamony, Madhan Karky Vairamuthu, and Maria E Orlowska. On optimal route of a calibrating mobile sink in a wireless sensor network. In *Networked Sensing Systems, 2007. INSS’07. Fourth International Conference on*, pages 61–64. IEEE, 2007.

- [83] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. Siam, 2002.
- [84] S Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Addison-Wesley, Reading, MA, 1990.
- [85] Dominik Schultes. Euclidean traveling salesman problem. 2004.
- [86] Sangit Chatterjee, Cecilia Carrera, and Lucy A Lynch. Genetic algorithms and traveling salesman problems. *European journal of operational research*, 93(3):490–510, 1996.
- [87] Iván Maza, Fernando Caballero, Jesús Capitán, JR Martínez-de Dios, and Aníbal Ollero. Experimental results in multi-uav coordination for disaster management and civil security applications. *Journal of intelligent & robotic systems*, 61(1-4):563–585, 2011.
- [88] Michael Hirsch and Daniel Schroeder. Dynamic decentralized cooperative control of multiple autonomous vehicles with multiple tasks for urban operations. In *AIAA Guidance, Navigation, and Control Conference*, 2012.
- [89] Ting Su and Jennifer G Dy. In search of deterministic methods for initializing k-means and gaussian mixture clustering. *Intelligent Data Analysis*, 11(4):319–338, 2007.
- [90] Stuart Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.
- [91] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- [92] Daniel L Alspach and Harold W Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *Automatic Control, IEEE Transactions on*, 17(4):439–448, 1972.
- [93] Avinash Kak. Expectation-maximization algorithm for clustering multidimensional numerical data. November 2014.
- [94] Frank Dellaert. The expectation maximization algorithm. 2002.
- [95] Sean Borman. The expectation maximization algorithm-a short tutorial, updated in 2009. *Submitted for publication*, pages 1–9, 2004.
- [96] HO Hartley. Maximum likelihood estimation from incomplete data. *Biometrics*, 14(2):174–194, 1958.

- [97] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [98] Tood K Moon. The expectation-maximization algorithm. *Signal processing magazine, IEEE*, 13(6):47–60, 1996.
- [99] Derek Kingston. personal communication, June 2013.
- [100] Charles R Hargraves and SW Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.
- [101] Michael A Hurni, Pooya Sekhavat, Mark Karpenko, and I Michael Ross. A pseudospectral optimal motion planner for autonomous unmanned vehicles. In *American Control Conference (ACC), 2010*, pages 1591–1598. IEEE, 2010.
- [102] Nathan E Smith, Richard G Cobb, Scott J Pierce, and Vincent M Raska. Optimal collision avoidance trajectories via direct orthogonal collocation for unmanned/remotely piloted aircraft sense and avoid operations. In *Proceedings of the AIAA Guidance, Navigation and Control Conference, National Harbor, Maryland*, 2014.
- [103] S.M. Ross. *Stochastic Real-Time Optimal Control: A Pseudospectral Approach for Bearing-Only Trajectory Optimization*. PhD thesis, Air Force Institute of Technology, 2011.
- [104] Brian R Geiger, Joseph F Horn, Gregory L Sinsley, James A Ross, Lyle N Long, and Albert F Niessner. Flight testing a real-time direct collocation path planner. *Journal of guidance, control, and dynamics*, 31(6):1575–1586, 2008.
- [105] Angela W Suplisson. *Optimal Recovery Trajectories for Automatic Ground Collision Avoidance Systems (Auto GCAS)*. PhD thesis, Air Force Institute of Technology, 2015.
- [106] Tadeusz J Masternak. *Multi-Objective Trajectory Optimization of a Hypersonic Reconnaissance Vehicle with Temperature Constraints*. PhD thesis, Air Force Institute of Technology, 2014.
- [107] Geoffrey T Huntington and Anil V Rao. Optimal reconfiguration of spacecraft formations using the gauss pseudospectral method. *Journal of Guidance, Control, and Dynamics*, 31(3):689–698, 2008.
- [108] John T Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.
- [109] D.E Kirk. *Optimal Control Theory, An Introduction*. Dover Publications Inc., 1998.

- [110] Geoffrey Todd Huntington. *Advancement and analysis of a Gauss pseudospectral transcription for optimal control problems*. PhD thesis, Citeseer, 2007.
- [111] David A Benson, Geoffrey T Huntington, Tom P Thorvaldsen, and Anil V Rao. Direct trajectory optimization and costate estimation via an orthogonal collocation method. *Journal of Guidance, Control, and Dynamics*, 29(6):1435–1440, 2006.
- [112] Divya Garg, Michael A Patterson, William W Hager, Anil V Rao, David A Benson, and Geoffrey T Huntington. An overview of three pseudospectral methods for the numerical solution of optimal control problems, 2009.
- [113] Anil V Rao. A survey of numerical methods for optimal control.
- [114] Steven M Ross, Richard G Cobb, William P Baker, and Frederick G Harmon. Implementation lessons and pitfalls for real-time optimal control with stochastic systems. *Optimal Control Applications and Methods*, 2014.
- [115] Ryan D. Gauntt. Aircraft course optimization tool using gops matlab code. No. afit/gse/env/12-m03, Air Force Institute of Technology, Wright Patterson AFB, Dayton, OH, March 2012.
- [116] Timothy R Jorris and Richard G Cobb. Multiple method 2-d trajectory optimization satisfying waypoints and no-fly zone constraints. *Journal of Guidance, Control, and Dynamics*, 31(3):543–553, 2008.
- [117] David Benson. *A Gauss pseudospectral transcription for optimal control*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [118] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.
- [119] Gary M. Goff. *Orbit Estimation of Non-Cooperative Maneuvering Spacecraft*. PhD thesis, Air Force Institute of Technology, 2015.
- [120] MATLAB. *version 8.3.0.532 (R2014a), 64-bit (maci64)*. The MathWorks Inc., Natick, Massachusetts, 2014.
- [121] John D’Errico. Interparc. <http://www.mathworks.com/matlabcentral/fileexchange/34874-interpac>.
- [122] David G Hull. Fundamentals of airplane flight mechanics, 2007.
- [123] Unmanned systems integrated roadmap fy 2011-2036. Number ADA558615, Washington DC, Oct 2011. Office of the Under Secretary of Defense (Acquisition Technology and Logistics).

- [124] Rao A. V. Patterson, M. A. *GPOPS-II Version 1.0: A General-Purpose MATLAB Toolbox for Solving Optimal Control Problems Using the Radau Pseudospectral Method*. University of Florida, Gainesville, FL 32611-6250, USA, 1 edition, January 2013.
- [125] Rao A. V. Patterson, M. A. Gpops-ii: A matlab software for solving multi-phase optimal control problems using hp-adaptive gaussian quadrature collocation method and sparse nonlinear programming. *ACM Transactions on Math. Software*, 39(3), Jul 13.
- [126] Hager W. W. Patterson, M. A. and A. V. Rao. A ph collocation scheme for optimal control. *Optimal Control Applications and Methods*, September 2013.
- [127] Matthew Duquette. The common mission automation services interface.
- [128] Nidal M. Jodeh and Richard Cobb. *Autonomous Flight Path Planning for Traffic Monitoring in Wireless Sensor Networks*. American Institute of Aeronautics and Astronautics, 2015/08/05 2015.

Vita

Lieutenant Colonel Nidal Jodeh is a PhD student studying aircraft control systems and optimization in the Department of Aeronautics and Astronautics at the Air Force Institute of Technology (AFIT), WPAFB, OH. Lt Col Jodeh was born and raised in Denver, Colorado and after graduating from Overland High School in 1993, he attended Colorado State University. He graduated in 1998 with a bachelor's degree in Engineering Science with a focus in Aerospace Engineering, and was recognized as a Distinguished Graduate of the Air Force Reserve Officer Training Corps.

In 2000, after initial training, Nidal was assigned to his first operational assignment with the 319th Missile Squadron at F.E. Warren AFB, Wyoming as an ICBM launch officer. While at F.E. Warren he served over 200 alerts as an ICBM Crew Commander, Instructor, and Alternate Command Post Commander, responsible for 150 on-alert ICBMs and over 100 on-alert officers and staff. He later served as a squadron flight commander and eventually the 90th Operations Group Executive Officer.

In 2004, he was accepted to the Air Force Institute of Technology at Wright Patterson AFB, in Dayton, Ohio. There, he began a Master of Science in Aeronautical Engineering, specializing in unmanned vehicle flight control, modeling, and simulation.

Upon graduation from the Air Force Institute of Technology in 2006 he was assigned to the Air Vehicles Control Sciences Branch at the Air Force Research Labs at Wright Patterson AFB. His programs researching single pilot, multi-vehicle control resulted in several publications and Air Force level awards. He later led development of technologies for autonomous reusable first stage booster systems.

In 2009, he moved to his new assignment at the US Air Force Academy, Colorado Springs, Colorado, where he developed and directed course curriculums for military education. Soon he was requested to transition from military training to academics and taught courses in the Fundamentals of Aeronautics and Aircraft Performance and Static Stability for the Department of Aeronautics.

After completing his PhD at AFIT, he was assigned to the Joint Strike Fighter Program Office.

| REPORT DOCUMENTATION PAGE | | | | | <i>Form Approved OMB No. 0704-0188</i> | |
|---|--------------------|-----------------------|-----------------------------------|---|--|--|
| The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS. | | | | | | |
| 1. REPORT DATE (DD-MM-YYYY) | | 2. REPORT TYPE | | 3. DATES COVERED (From — To) | | |
| 26 – 12 – 2015 | | Doctoral Dissertation | | Aug 2012 — Dec 2015 | | |
| 4. TITLE AND SUBTITLE | | | | 5a. CONTRACT NUMBER | | |
| Optimal UAS Assignments and Trajectories for Persistent Surveillance and Data Collection from a Wireless Sensor Network | | | | 5b. GRANT NUMBER | | |
| | | | | 5c. PROGRAM ELEMENT NUMBER | | |
| 6. AUTHOR(S) | | | | 5d. PROJECT NUMBER | | |
| Nidal M. Jodeh | | | | 5e. TASK NUMBER | | |
| | | | | 5f. WORK UNIT NUMBER | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) | | | | 8. PERFORMING ORGANIZATION REPORT NUMBER | | |
| Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765 | | | | AFIT-ENY-DS-15-S-062 | | |
| 9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) | | | | 10. SPONSOR/MONITOR'S ACRONYM(S) | | |
| Air Force Research Laboratory, Aerospace Systems Directorate (AFRL/RQQA) Attn: Dr. Derek Kingston 2210 8th Street, Bldg. 146, Rm. 301 WPAFB, OH, 45433 DSN 713-7038, COMM 937-255-6301 Email: derek.kingston@us.af.mil | | | | AFRL/RQQA | | |
| 12. DISTRIBUTION / AVAILABILITY STATEMENT | | | | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) | | |
| DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. | | | | | | |
| 13. SUPPLEMENTARY NOTES | | | | | | |
| This material is declared a work of the U.S. Government as is not subject to copyright protection in the United States. | | | | | | |
| 14. ABSTRACT | | | | | | |
| This research developed a method for multiple Unmanned Aircraft Systems (UAS) to efficiently collect data from a Wireless Sensor Networks (WSN). WSN are composed of any number of fixed, ground-based sensors that collect and upload local environmental data to overflying UAS. The three-step method first uniquely assigns aircraft to specific sensors on the ground. Second, an efficient flight path is calculated to minimize the aircraft flight time required to verify their assigned sensors. Finally, sensors reporting relatively higher rates of local environmental activity are re-assigned to dedicated aircraft tasked with concentrating on only those sensors. This work was sponsored by the Air Force Research Laboratory, Control Sciences branch, at Wright Patterson AFB. Based on simulated scenarios and preliminary flight tests, optimal flight paths resulted in a 14% to 32% reduction in flight time and distance when compared to traditional flight planning methods. | | | | | | |
| 15. SUBJECT TERMS | | | | | | |
| Wireless Sensor Networks (WSN), Unmanned Aircraft Systems (UAS), Mobile Agent, Optimal Trajectories, Trajectory, Data Collection, Multiple UAS, Assignment, k-means, Multi-Variate Gaussian Mixture Models, Expectation Maximization (EM), Direct Orthogonal Collocation, Optimal Control, Optimization, Pseudospectral Method, Flight Test | | | | | | |
| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | | 18. NUMBER OF PAGES | |
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | 19a. NAME OF RESPONSIBLE PERSON | |
| U | U | U | U | | Dr. Richard G. Cobb, AFIT/ENY | |
| | | | | | 19b. TELEPHONE NUMBER (include area code) | |
| | | | | | (937) 255-3636, x4559; richard.cobb@afit.edu | |